Digital Imaging and Communications in Medicine (DICOM)

Sup 243 - Label Map Segmentation

DICOM Standards Committee - Working Group 6 - Base Standard

1300 N. 17th Street Suite 1752
Rosslyn
VA
22209
USA

Status: Public Comment
Publication date 2024/01/08

This supplement is prepared pursuant to work item: 2023-10-B

Table of Contents

List of Tables

Document History

Document Version	Date	Content
01	2023/11/08	First draft for review by WG 6
02	2023/11/09	Include WG 6 first read feedback
03	2024/01/07	Draft for review by WG-06 for public comment
04	2024/01/08	Public comment

To Do After Public Comment

1	? hyphenate "8-bit" everywhere?
2	Insert definitions for things like Photometric Interpretation in Segmentation Image Module editorially.
3	Consider factoring out the bits allocated stuff into a separate section or table as for some other images IODs. Also, should the HighBit be predicated on BitsStored or BitsAllocated?
4	Add sub-headings for each type to C.8.20.2.3.
5	Consider extending PS3.17 Annex HH example to describe differences in LABELMAP attribute encoding +/- picture comparing them.

Open Issues

2	The proposal is designed to support segmenting classes of structure, and also adequately supports small to moderate numbers of instances of a single class (with great redundancy of coded description of such instances, since the category/property needs to be repeated for every numbered segment). There is no mechanism for efficiently describing very large number of instances of a class segmentation, and to do so we would probably need to re-factor the Segment Description Macro to use the category/property for the class, and specify that it applied to multiple instances identified by the encoded index value. Is this a reasonable decision to restrict the scope in the interest of simplicty of the most common use case?
3	Is 16 bits per pixel a sufficient number of indices? For class rather than instance segmentation, it would seem to be, but if we were to explicitly try to support very large numbers of instances, we might need 32 bits. 8 bits is often sufficient but definitely too few for some applications (e.g., anatomic structure atlases), hence the decision to allow either 8 or 16. Going beyond 16 bits would require a change to the Palette Color architecture, if that were to be required, though it might not be appropriate for that use case (hard to imagine that many discrete colors being useful).
4	Is it sufficient to relax the sequential numbering constraint for LABELMAP types, but retain it for existing BINARY and FRACTIONAL? In the existing IOD, Segment Number is constrained to start at a value of 1, and increase monotonically by 1. This is impractical for label maps in common use which are commonly defined across projects, atlases or some other set of instances that may encode only some of the segments defined in one instance, and these may start from 0, or sparsely use the range of numbers available. It is desirable to be able to literally use the existing label map indices in other formats when transcoding. Alternatively, the (rather pointless) constraint in the existing IOD could be completely removed, but that might affect receiving implementations that depend on it, if any.
5	Given that a labelmap may contain many segments, and each pixel value is an index, it makes sense to allow using the existing Palette Color LUT mechanism that is used in ultrasound, etc., which supports the use of 8 or 16 bit indices, the use of 8 or 16 bit color entries (theoretically), and which implies use of the dedicated PALETTE COLOR Photometric Interpretation, and irrelevance of any color specified within each Segment Description Item, and the use of an ICC Profile to achive color consistency. Should we constain the color entries to be only 8 bits (as is done for standalone Color Palette IOD), or only 16 bits (as is done for Image or Presentation State IOD, and hence applies to the modified Segmentation IOD, which is an "Image"), or allow for both? A well-known UID may be specified, but the LUT is still required to be encoded with each instance. Should we allow for a referenced palette without inclusion of its data or is support of standalone Palette Color Storage SOP Class (a non-Patient-related SOP Class) too poor at this time for that to be practical? The use of the so-called "Segmented" definition of the color tables (by equation rather than literally, not to be confused with Segmentation of images) is permitted, though unlikely to be used for this application, and since it is not widely implemented, should it be forbidden?
6	For multiple labelmaps in separate instances applying to the same images, is there a need (beyond say grouping them in a Series) to signal that they are in some way related, if they are, or does commonality of other attribute values (such as those that are creator-related) suffice? E.g., the current supplement allows for describing segments that are not actually used in the instance and may be common to other instances, so is there a need to identify those that share the same segments (although the case where the segment descripion is re-usable rather than specific would need to be distinguished)? One possibility would be to state that one labelmap (in one instance) is a sub-class of a segment in a parent labelmap (in another instance), e.g., parts of a tumor in a tumor class.
7	When converting from (non-overlapping) binary to labelmap representation or vice versa, does the predecessor need to be referenced and if so, how (in a manner that is distinct from referencing the images that were segmented in the first place)?
8	Should both MONOCHROME2 and PALETTE COLOR Photometric Interpretation be allowed when Segmentation Type is LABELMAP, e.g., when no colors are suggested/assigned or only the latter?

9	The matter of having to describe every pixel value implying that the value 0 will also be described (which might serve as the background) is in conflict with the existing requirement to start Segment Number at 1 not zero. Should the existing (somewhat gratuitous) requirement be relaxed or would that threaten the installed base of valid implementations? Could Pixel Padding Value or a similar mechanism be re-used in some way to avoid the need to describe a pixel value of zero as a Segment? What would this mean for conversions to/from BINARY form? (This would need a change to the constraint in PS3.3 A.51.4).
10	What constraints (if any) should be placed on Segment Number when Segmentation Type is LABELMAP, if any? It is a goal of the supplement to allow for conversion of existing labelmap pixel data arrays in other formats without renumbering, so presumably none (other than those implied by bit depth limits). Limits on the potential values are distinct from limits on the ordering (e.g., ascending by numeric value) in the description (Item order) in the Segment Sequence (which is currently not actually specified but might be implied by the ascending value; this is made explicit in notes).

Closed Issues

1	The existing IOD is modified and two SOP Classes are defined in the same manner as used for Digital X-Ray for processing and for presentation, and conditions specified in the IOD and included Modules as necessary, based on the SOP Class or the Segmentation Type (in the same manner as Presentation Intent Type for DX). Alternatively, two separate IODs could be defined, but there would be a lot of repetition.
2	There is no mechanism to specify that different ranges of bits within the pixel data value (index) are used to specify different things (e.g., class vs. instance, different classes, or overlap of classes).
3	Only a single label map is allowed in each SOP Instance. Overlap of different segments can be addressed by encoding multiple label maps in separate SOP Instances.
4	The size of the Per-Frame Functional Group Sequence (in which, for every frame, geometry is specified) are not addressed in this Supplement, except to the extent that for a label map, the Segmentation Macro is not required since the referenced segment is specified by the pixel value instead.

Scope and Field of Application

This Supplement describes addition of a Label Map Segmentation IOD to DICOM to encode classification of entities.

Currently, the DICOM standard supports an IOD and SOP Class for pixel- or voxel-based segmentation encoding, in which each segmented property is represented as a binary bit plane (or an 8 bit probabilistic or occupancy value). While this allows for overlapping of segments, it is inefficient and difficult to encode large numbers of non-overlapping segmentations, as they require non-trivial processing both to extract from the bit plane encoded data, to assure there is no overlap, and to convert to the label map form that is very commonly used internally and persistently for clinical applications.

The current DICOM bit-plane-based segmentation methods have proven to be awkward both for 3D cross-sectional imaging applications when there are very large numbers of slices and/or structures, and for whole slide microscopy imaging, when there are very large numbers of tiles and/or property classes. They are also typically large and sparse and should compress well but there are very few single bit compression schemes supported by the standard and they do not do well with these types of images.

This Supplement defines a label map segmentation enhanced multi-frame IOD that specifies a data structure that provides, for each pixel or voxel in 2D, 3D or tiled pyramidal space, an index value conveying the non-overlapping segment for each pixel. Existing data elements for describing segmentations are reused where appropriate. Bit depth is sufficient (8, 16) to encode large numbers of segments but allow for more compact encoding. The existing palette color photometric interpretation may be used (instead of monochrome) if colors are to be suggested, to leverage the widespread implementations in toolkits, and to allow for the use of existing lossless compression schemes. Segment properties are conveyed in the existing segment description structure so as to be compatible with the existing bit plane segment descriptions. Re-using the segment description does not prevent the use of separately encoded or well-known DICOM color palette objects.

The scope is confined to label maps for "classes" (what "class" a segment represents) but not "instances' (which "instance" of a "class" is represented), where classes and instances are separately communicated by the pixel value (e.g., if one wants to individually identify nuclei rather than treat them all as being of one class). This might be the subject of a future extension.

The scope is confined to a single label map, which does not allow for overlap of different segments. If overlapping of multiple label maps is required, separate SOP Instances may be created.

Issues related to the efficient representation (or avoidance) of the Per-Frame Functional Group Sequence (in which, for every frame, the Referenced Segment Number is specified) are out of scope, and may be addressed in a separate Supplement or CP if necessary.

PS3.3 DICOM PS3.3 - Information Object Definitions

*Amend PS3.3 as follows (changes to existing text are bold and **underlined** for additions and **~~struckthrough~~** for removals):*

A.1.4 Overview of the Composite IOD Module Content

Table A.1-1c. Composite Information Object Modules Overview - More Images

IODs **Modules**	**Seg**	**Label Map Seg**
Patient	M	**M**
Clinical Trial Subject	U	**U**
General Study	M	**M**
Patient Study	U	**U**
Clinical Trial Study	U	**U**
General Series	M	**M**
Clinical Trial Series	U	**U**
Segmentation Series	M	**M**
Frame of Reference	C	**C**
General Equipment	M	**M**
Enhanced General Equipment	M	**M**
Acquisition	U	**U**
Multi-Resolution Pyramid	U	**U**
General Image	M	**M**
General Reference	U	**U**
Microscope Slide Layer Tile Organization	C	**C**
Image Pixel	M	**M**
Palette Color Lookup Table		**C**
Multi-frame Functional Groups	M	**M**
Multi-frame Dimension	M	**M**
Specimen	U	**U**
Segmentation Image	M	**M**
Common Instance Reference	C	**C**
ICC Profile		**C**
SOP Common	M	**M**
Frame Extraction	C	**C**

A.51 Segmentation IOD

A.51.1 Segmentation IOD Description

The Segmentation IOD specifies a multi-frame image representing a classification of pixels in one or more referenced images. Segmentations are either binary**~~, or~~** fractional**, or label map**. If the referenced images have a defined Frame of Reference, the Segment-

ation Instance shall have the same Frame of Reference and is not required to have the same spatial sampling or extent as the referenced images. If the referenced image does not have a defined Frame of Reference, the Segmentation Instance shall have the same spatial sampling and extent as the referenced image.

The Segmentation IOD does not include the full set of acquisition parameters of the referenced images, e.g., cardiac phase. An application rendering or processing the segmentation may need to access the referenced images for such information.

The Segmentation IOD is used in two SOP Classes as defined in PS3.4 Storage Service Class, a SOP Class for storage of binary and fractional segmentations, and a SOP Class for storage of label map segmentations. These are distinguished by their SOP Class UID and by the Enumerated Value of the mandatory Attribute Segmentation Type (0062,0001) in the Segmentation.

A.51.2 Segmentation IOD Entity-Relationship Model

This IOD uses the E-R Model in ???, with only the Image IE below the Series IE.

A.51.3 Segmentation IOD Module Table

Table A.51-1 specifies the Modules of the Segmentation IOD.

Table A.51-1. Segmentation IOD Modules

IE	Module	Reference	Usage
Patient	Patient	???	M
	Clinical Trial Subject	???	U
Study	General Study	???	M
	Patient Study	???	U
	Clinical Trial Study	???	U
Series	General Series	???	M
	Segmentation Series	C.8.20.1	M
	Clinical Trial Series	???	U
Frame of Reference	Frame of Reference	???	C - Required if Derivation Image Functional Group (???) is not present. May be present otherwise.
Equipment	General Equipment	???	M
	Enhanced General Equipment	???	M
Acquisition	General Acquisition	???	U
Multi-Resolution Pyramid	Multi-Resolution Pyramid	???	U
Image	General Image	???	M
	General Reference	???	U
	Microscope Slide Layer Tile Organization	???	C - Required if Dimension Organization Type (0020,9311) is present with a value of TILED_FULL. May be present otherwise.
	Image Pixel	???	M
	Segmentation Image	C.8.20.2	M
	Multi-frame Functional Groups	???	M
	Multi-frame Dimension	???	M
	Palette Color Lookup Table	**C.7.9**	**C - Required if Photometric Interpretation (0028,0004) has a value of PALETTE COLOR**

IE	Module	Reference	Usage
	Specimen	???	U
	Common Instance Reference	???	C - Required if Derivation Image Functional Group (???) is present.
	ICC Profile	**???**	**C - Required if Photometric Interpretation (0028,0004) has a value of PALETTE COLOR**
	SOP Common	???	M
	Frame Extraction	???	C - Required if the SOP Instance was created in response to a Frame-Level retrieve request

Note

The ??? was previously included in this IOD but has been retired, and its functionality replaced by the ???. See PS3.3-2008.

A.51.4 Segmentation IOD Content Constraints

The VOI LUT Module shall not be present.

The Modality LUT Module shall not be present.

The Overlay Plane Module shall not be present.

Pixel Padding Value (0028,0120) shall not be present.

For Purpose of Reference Code Sequence (0040,A170) within Source Instance Sequence (0042,0013) in the ??? DCID 7019 “Segmentation Non-Image Source Purpose of Reference” shall be used.

A.51.5 Segmentation Functional Groups

Table A.51-2 specifies the use of the Functional Group Macros used in the ??? for the Segmentation IOD.

Table A.51-2. Segmentation Functional Group Macros

Functional Group Macro	Section	Usage
Pixel Measures	???	C - Required if Derivation Image Functional Group (???) is not present and the Frame of Reference is defined in the patient-relative Reference Coordinate System. May be present otherwise if the Frame of Reference is defined in the patient-relative Reference Coordinate System. See Section A.51.5.1
Plane Position (Patient)	???	C - Required if Derivation Image Functional Group (???) is not present and the Frame of Reference is defined in the patient-relative Reference Coordinate System. May be present otherwise if the Frame of Reference is defined in the patient-relative Reference Coordinate System. See Section A.51.5.1
Plane Orientation (Patient)	???	C - Required if Derivation Image Functional Group (???) is not present and the Frame of Reference is defined in the patient-relative Reference Coordinate System. May be present otherwise if the Frame of Reference is defined in the patient-relative Reference Coordinate System. See Section A.51.5.1
Plane Position (Slide)	???	C - Required if Derivation Image Functional Group (???) is not present and the Frame of Reference is defined in the Slide Coordinate System and Dimension Organization Type (0020,9311) is not TILED_FULL. May be present otherwise if the Frame of Reference is defined in the Slide Coordinate System. See Section A.51.5.1.
Derivation Image	???	C - Required if Pixel Measures (???) or either Plane Position (Patient) (???) or Plane Orientation (Patient) (???) (if the Frame of Reference is defined in the patient-relative Reference Coordinate System), or Plane Position (Slide) (???) (if the Frame of Reference is defined in the Slide Coordinate System) Functional Groups are not present. May be present otherwise. See Section A.51.5.1

Functional Group Macro	Section	Usage
Frame Content	???	M
Segmentation	C.8.20.3.1	~~M~~C - Required if Segmentation Type (0062,0001) is not LABELMAP.

A.51.5.1 Segmentation Functional Groups Description

When a Frame of Reference UID is present the segment shall be specified within that coordinate system, using the Pixel Measures and either the Plane Position (Patient) and Plane Orientation (Patient), or the Plane Position (Slide) Functional Groups. Since this defines the spatial relationship of the segment, the size of the segmentation frames need not be the same size, or resolution, as the image data used to generate the segment data. The Derivation Image Functional Group may also be present, to specify on which images the segmentation was actually performed (since there may be others in the same Frame of Reference that are spatially co-located, but were not used to perform the segmentation).

If the Frame of Reference UID is not present, each pixel of the segmentation shall correspond to a pixel in a referenced image, using the Derivation Image Functional Group. Hence, the rows and columns of each referenced image will match the segmentation image. If both the Frame of Reference UID and the Derivation Image Functional Group are present, the segmentation and referenced image pixels need not correspond.

The value of Purpose of Reference Code Sequence (0040,A170) in the ??? shall be (121322, DCM, "Source Image for Image Processing Operation"). The value of Derivation Code Sequence (0008,9215) shall be (113076, DCM, "Segmentation").

Note

Non-image source Instances used during segmentation, such as Real World Value maps, can be described in the top level Data Set in the Source Instance Sequence (0042,0013) of the ??? and are implied to have been used for the derivation of all frames. I.e., there is no mechanism for selectively specifying on a per-frame basis which non-Image Instances were used. Real World Value Map Instances already contain a means of selectively applying different scale factors to different frames.

C.8.20 Segmentation

This section describes the specific Modules for the Segmentation IOD.

C.8.20.1 Segmentation Series Module

Table C.8.20-1 specifies the Attributes of the Segmentation Series Module.

Table C.8.20-1. Segmentation Series Module Attributes

Attribute Name	Tag	Type	Attribute Description
Modality	(0008,0060)	1	Type of device, process or method that created the Instances in this Series **Enumerated Values:** **SEG**
Series Number	(0020,0011)	1	A number that identifies this Series.
Referenced Performed Procedure Step Sequence	(0008,1111)	1C	Uniquely identifies the Performed Procedure Step SOP Instance to which the Series is related Only a single Item shall be included in this Sequence. Required if a Performed Procedure Step SOP Class was involved in the creation of this Series.
>Include ???			

C.8.20.2 Segmentation Image Module

Table C.8.20-2 specifies the Attributes of the Segmentation Image Module.

Table C.8.20-2. Segmentation Image Module Attributes

Attribute Name	Tag	Type	Attribute Description
Image Type	(0008,0008)	1	Value 1 shall be DERIVED. Value 2 shall be PRIMARY. No other values shall be present.
Include ???			
Samples Per Pixel	(0028,0002)	1	**Enumerated Values:** **1**
Photometric Interpretation	(0028,0004)	1	**Enumerated Values <u>if Segmentation Type (0062,0001) is BINARY or FRACTIONAL</u>:** **MONOCHROME2** **<u>Enumerated Values if Segmentation Type (0062,0001) is LABELMAP</u>:** **<u>MONOCHROME2</u>** **<u>PALETTE COLOR</u>**
Pixel Representation	(0028,0103)	1	**Enumerated Values:** **0**
Bits Allocated	(0028,0100)	1	See Section C.8.20.2.1. **Enumerated Values if Segmentation Type (0062,0001) is BINARY:** **1** **Enumerated Values if Segmentation Type (0062,0001) is ~~not BINARY~~<u>FRACTIONAL</u>:** **8** **<u>Enumerated Values if Segmentation Type (0062,0001) is LABELMAP</u>:** **<u>8</u>** **<u>16</u>**

Attribute Name	Tag	Type	Attribute Description
Bits Stored	(0028,0101)	1	See Section C.8.20.2.1. **Enumerated Values if Segmentation Type (0062,0001) is BINARY:** **1** **Enumerated Values if Segmentation Type (0062,0001) is ~~not BINARY~~FRACTIONAL:** **8** **Enumerated Values if Segmentation Type (0062,0001) is LABELMAP and Bits Allocated (0028,0100) is 8:** **8** **Enumerated Values if Segmentation Type (0062,0001) is LABELMAP and Bits Allocated (0028,0100) is 16:** **16**
High Bit	(0028,0102)	1	See Section C.8.20.2.1. **Enumerated Values if Segmentation Type (0062,0001) is BINARY:** **0** **Enumerated Values if Segmentation Type (0062,0001) is ~~not BINARY~~FRACTIONAL:** **7** **Enumerated Values if Segmentation Type (0062,0001) is LABELMAP and Bits Allocated (0028,0100) is 8:** **7** **Enumerated Values if Segmentation Type (0062,0001) is LABELMAP and Bits Allocated (0028,0100) is 16:** **15**
Lossy Image Compression	(0028,2110)	1	Specifies whether an Image has undergone lossy compression (at a point in its lifetime), or is derived from lossy compressed images. **Enumerated Values:** **00** Image has NOT been subjected to lossy compression. **01** Image has been subjected to lossy compression. Once this value has been set to "01" it shall not be reset. See Section C.8.20.2.2 and ???.

Attribute Name	Tag	Type	Attribute Description
Lossy Image Compression Ratio	(0028,2112)	1C	Describes the approximate lossy compression ratio(s) that have been applied to this image. See ???. Required if present in the source images or this IOD Instance has been compressed.
Lossy Image Compression Method	(0028,2114)	1C	A label for the lossy compression method(s) that have been applied to this image. See ???. Required if present in the source images or this IOD Instance has been compressed. See Section C.8.20.2.2.
Segmentation Type	(0062,0001)	1	The type of encoding used to indicate the presence of the segmented property at a pixel/voxel location. **Enumerated Values:** **BINARY** **FRACTIONAL** **LABELMAP** See Section C.8.20.2.3.
Segmentation Fractional Type	(0062,0010)	1C	For fractional segmentation encoding, the meaning of the fractional value. Required if Segmentation Type (0062,0001) is FRACTIONAL. See Section C.8.20.2.3 for Enumerated Values.
Maximum Fractional Value	(0062,000E)	1C	Specifies the value that represents a probability of 1 or complete occupancy. There shall be no values in Pixel Data (7FE0,0010) greater than this value. Required if Segmentation Type (0062,0001) is FRACTIONAL.
Segments Overlap	(0062,0013)	3	Whether or not any segments in this Instance overlap. I.e., whether or not any pixel is or might be in more than one segment. **Enumerated Values:** **YES** Some segments overlap **UNDEFINED** Some segments might overlap **NO** No segments overlap See Section C.8.20.2.3. **If present, shall be NO if Segmentation Type (0062,0001) is LABELMAP.** **Note** If the value is NO, then a receiving application to which this matters can be assured that no segments overlap and does not need to check every pixel. If the value is UNDEFINED or YES, or the Attribute is absent, then a receiving application might need to check every pixel in every segment.

Attribute Name	Tag	Type	Attribute Description
Segment Sequence	(0062,0002)	1	Describes the segments that are contained within the data. One or more Items shall be included in this Sequence. **Note** **The Items of this Sequence are not required to be in any particular order, i.e., are not required to be ordered by Segment Number (0062,0004).**
>Include Table C.8.20-4 "Segment Description Macro Attributes"			
>Segment Algorithm Name	(0062,0009)	1C	The name(s) of algorithm(s) used to generate the segment. Required if Segment Algorithm Type (0062,0008) is not MANUAL.
>Segmentation Algorithm Identification Sequence	(0062,0007)	3	A description of how this segment was derived. Algorithm Name (0066,0036) within this Sequence may be identical to Segment Algorithm Name (0062,0009). One or more Items are permitted in this Sequence. **Note** Previously, the Segment Surface Generation Algorithm Identification Code Sequence (0066,002D) was used, but it has been replaced in this Module, since not all segmentation algorithms involve surface generation. See PS3.3-2016d.
>>Include ???			*BCID 7162 "Surface Processing Algorithm Family".*
>Recommended Display Grayscale Value	(0062,000C)	3	A default single gray unsigned value in which it is recommended that the maximum pixel value in this segment be rendered on a monochrome display. The units are specified in P-Values from a minimum of 0000H (black) up to a maximum of FFFFH (white). **Note** The maximum P-Value for this Attribute may be different from the maximum P-Value from the output of the Presentation LUT, which may be less than 16 bits in depth.
>Recommended Display CIELab Value	(0062,000D)	3	A default triplet value in which it is recommended that segment be rendered on a color display. The units are specified in PCS-Values, and the value is encoded as CIELab. See ???. **Shall not be present if Segmentation Type (0062,0001) is LABELMAP and Photometric Interpretation (0028,0004) is PALETTE COLOR.**

C.8.20.2.1 Bits Allocated and Bits Stored

As a consequence of the enumerated Bits Allocated and Bits Stored Attribute values, single bit pixels shall be packed 8 to a byte as defined by the encoding rules in PS3.5.

C.8.20.2.2 Lossy Image Compression and Lossy Image Compression Method

If Lossy Image Compression (0028,2110) in any of the source images is "01", the value shall be "01" for the Segmentation Instance.

The process of segmentation itself is defined not to be lossy compression, even though it involves loss. If the Segmentation Instance is encoded using a lossy compression Transfer Syntax, then the value shall be set to "01".

Note

It is not advisable to lossy compress a Segmentation SOP Instance. In particular, a binary segmentation should not be lossy compressed.

C.8.20.2.3 Segmentation Type, Segmentation Fractional Type and Segments Overlap

A Segmentation Type (0062,0001) of BINARY indicates the segmented property is present with a value of 1 and absent with a value of 0.

For a Segmentation Type (0062,0001) of FRACTIONAL the segmented property is defined as a value from zero to the Maximum Fractional Value (0062,000E). A FRACTIONAL segmentation shall be further specified via Segmentation Fractional Type (0062,0010).

Enumerated Values of Segmentation Fractional Type (0062,0010):

PROBABILITY Defines the probability, as a ratio of the pixel value to the Maximum Fractional Value, that the segmented property occupies the spatial area defined by the voxel.

OCCUPANCY Defines the proportion of the pixel volume occupied by the segmented property as the ratio of the pixel value to the Maximum Fractional Value.

A Segmentation Type (0062,0001) of LABELMAP indicates the presence of the segmented property is conveyed by the value of the Pixel Data (7FE0,0010), which is one of the values of Segment Number (0062,0004). In a label map, each pixel contains a coded value that indicates the segment, rather than the encoding of a separate bit plane for each segment.

Note

Binary Segmentation Instances may be transformed into or from **"**label maps**"**, ~~**in which each pixel contains a coded value that indicates the segment, rather than the Segmentation IOD representation that encodes separate bit planes for each segment**~~. This conversion is facilitated by knowing whether or not any segment bitplanes overlap. **By definition, each pixel value in a single Label Map Segmentation Instance can only represent one segment. and cannot overlap.** A Segments Overlap (0062,0013) value of NO indicates they can be converted into (or may have been converted from) a label map representation without the need to check every pixel.

For a Segmentation Type (0062,0001) of LABELMAP, every pixel value actually encoded in Pixel Data (7FE0,0010) is required to be described in an Item of Segment Sequence (0062,0002).

Note

1. **The converse is not true, in that the Segment Sequence (0062,0002) can describe segments that are not actually present in the pixel data, e.g., to allow for re-use of a common segment description across multiple instances, despite the inefficiency of encoding unused information.**

2. **The need to describe every pixel value implies that the value 0 will also be described, which might serve as the background.**

C.8.20.2.4 Segment Number

Segment Number (0062,0004) shall be unique within each Instance**.**

If Segmentation Type (0062,0001) is BINARY or FRACTIONAL, **Segment Number (0062,0004) shall** start at a value of 1, and increase monotonically by 1.

Note

1. **When converting from LABELMAP Segmentation Instances to BINARY Segmentation Instances, care should be taken to reassign the Segment Number values if necessary, such that the requirements in this section are complied with.**

2. **Requirements on the ascending order of the Segment Number values does not imply an ordering of the Sequence Items of any enclosing Sequence.**

C.8.20.3 Segmentation Functional Group Macros

The following sections contain Functional Group Macros specific to the Segmentation IOD.

Note

The Attribute descriptions in the Functional Group Macros are written as if they were applicable to a single frame (i.e., the Macro is part of the Per-Frame Functional Groups Sequence). If an Attribute is applicable to all frames (i.e., the Macro is part of the Shared Functional Groups Sequence) the phrase "this frame" in the Attribute description shall be interpreted to mean "for all frames".

C.8.20.3.1 Segmentation Macro

Table C.8.20-3 specifies the Attributes of the Segmentation Macro.

Table C.8.20-3. Segmentation Macro Attributes

Attribute Name	Tag	Type	Attribute Description
Segment Identification Sequence	(0062,000A)	1	Identifies the characteristics of this frame. Only a single Item shall be included in this Sequence.
>Referenced Segment Number	(0062,000B)	1	Uniquely identifies the segment described in Segment Sequence (0062,0002) by reference to Segment Number (0062,0004). Referenced Segment Number (0062,000B) shall not be multi-valued.

C.8.20.4 Segmentation Macros

The following sections contain Macros specific to the Segmentation IOD.

C.8.20.4.1 Segment Description Macro

Table C.8.20-4 specifies the Attributes of the Segment Description Macro.

Table C.8.20-4. Segment Description Macro Attributes

Attribute Name	Tag	Type	Attribute Description
Segment Number	(0062,0004)	1	Identification number of the segment. The value of Segment Number (0062,0004) shall be unique within the Segmentation Instance in which it is created. See Section C.8.20.2.4.
Segment Label	(0062,0005)	1	User-defined label identifying this segment. This may be the same as Code Meaning (0008,0104) of Segmented Property Type Code Sequence (0062,000F).
Segment Description	(0062,0006)	3	User-defined description for this segment.
Segment Algorithm Type	(0062,0008)	1	Type of algorithm used to generate the segment. **Enumerated Values:** **AUTOMATIC** calculated segment **SEMIAUTOMATIC** calculated segment with user assistance **MANUAL** user-entered segment

Attribute Name	Tag	Type	Attribute Description
Include Table 10-7b "Multiple Site General Anatomy Optional Macro Attributes"			*May not be necessary if the anatomy is implicit in the Segmented Property Type Code Sequence.* *More than one Item in Anatomic Region Sequence (0008,2218) may be used when a region of interest spans multiple anatomical locations and there is not a single pre-coordinated code describing the combination of locations. There is no requirement that the multiple locations be contiguous.*
Segmented Property Category Code Sequence	(0062,0003)	1	Sequence defining the general category of the property the segment represents. Only a single Item shall be included in this Sequence.
>Include Table 8.8-1 "Code Sequence Macro Attributes"			*BCID 7150 "Segmentation Property Category".*
Segmented Property Type Code Sequence	(0062,000F)	1	Sequence defining the specific property the segment represents. **Note** "Property" is used in the sense of meaning "what the segmented voxels represent", whether it be a physical or biological object, be real or conceptual, having spatial, temporal or functional extent or not. I.e., it is what the segment "is" (as opposed to some feature, attribute, quality, or characteristic of it, like color or shape or size). Only a single Item shall be included in this Sequence.
>Include Table 8.8-1 "Code Sequence Macro Attributes"			*BCID 7151 "Segmentation Property Type".*
>Segmented Property Type Modifier Code Sequence	(0062,0011)	3	Sequence defining the modifier of the property type of this segment. One or more Items are permitted in this Sequence.
>>Include Table 8.8-1 "Code Sequence Macro Attributes"			*DCID 244 "Laterality".* **Note** *For Retinal Segmentation Surfaces, laterality is not typically specified.*
Tracking ID	(0062,0020)	1C	A text label used for tracking a finding or feature, potentially across multiple reporting objects, over time. This label shall be unique within the domain in which it is used. Required if Tracking UID (0062,0021) is present. **Note** 1. May or may not have the same value as Segment Label (0062,0005). 2. Related SR Instances may exist, for example, to record measurements related to this segment, but need not exist for this Attribute to be used. 3. This Attribute will have the same value as the value of the (112039, DCM, "Tracking Identifier") Content Item in SR Instances that reference this Segment in this Segmentation Instance.

Attribute Name	Tag	Type	Attribute Description
Tracking UID	(0062,0021)	1C	A unique identifier used for tracking a finding or feature, potentially across multiple reporting objects, over time. Required if Tracking ID (0062,0020) is present. **Note** 1. Related SR Instances may exist, for example, to record measurements related to this segment, but need not exist for this Attribute to be used. 2. This Attribute will have the same value as the value of the (112040, DCM, "Tracking Unique Identifier") Content Item in SR Instances that reference this Segment in this Segmentation Instance.
Definition Source Sequence	(0008,1156)	3	Instances containing the source of the Segment information. Only a single Item is permitted in this Sequence.
>Include Table 10-11 "SOP Instance Reference Macro Attributes".			
>Referenced ROI Number	(3006,0084)	1C	The value of ROI Number (3006,0022) in the referenced SOP Instance that identifies the ROI that is the origin of the Segment information. Required if Referenced SOP Class UID (0008,1150) is RT Structure Set Storage ("1.2.840.10008.5.1.4.1.1.481.3").
Include Table 10.9.3-1 "Content Creator Macro"			

C.7.9 Palette Color Lookup Table Module

Table C.7-22 specifies the Attributes of the Palette Color Lookup Table Module, which describe the Lookup table data for images with Palette Color photometric interpretation.

When the Palette Color Lookup Table Module is present in an Image IOD, the conditional requirements for the use of Palette Color Lookup Table Data (0028,1201-1203) and Segmented Palette Color Lookup Table Data (0028,1221-1223), described in Table C.7-22, shall take precedence over the conditional requirements described in the ??? (see ???). When the Palette Color Lookup Table Module is present in a Presentation State IOD, the Palette Color Lookup Table Data (0028,1201-1203) Attributes are mandatory and the Segmented Palette Color Lookup Table Data (0028,1221-1223) shall not be present. When the Palette Color Lookup Table Module is present in a Color Palette IOD, either the Palette Color Lookup Table Data (0028,1201-1203) or Segmented Palette Color Lookup Table Data (0028,1221-1223) Attributes may be used.

When the Palette Color Lookup Table Module is present in a Color Palette IOD, the 3rd value of Palette Color Lookup Table Descriptor (0028,1101-1103) (i.e, the number of bits for each entry in the Lookup Table Data) shall be 8.

Table C.7-22. Palette Color Lookup Table Module Attributes

Attribute Name	Tag	Type	Attribute Description
Include Table C.7-22a "Palette Color Lookup Table Macro Attributes"			

Table C.7-22a. Palette Color Lookup Table Macro Attributes

Attribute Name	Tag	Type	Attribute Description
Red Palette Color Lookup Table Descriptor	(0028,1101)	1	Specifies the format of the Red Palette Color Lookup Table Data (0028,1201). See Section C.7.6.3.1.5 for further explanation.
Green Palette Color Lookup Table Descriptor	(0028,1102)	1	Specifies the format of the Green Palette Color Lookup Table Data (0028,1202). See Section C.7.6.3.1.5 for further explanation.

Attribute Name	Tag	Type	Attribute Description
Blue Palette Color Lookup Table Descriptor	(0028,1103)	1	Specifies the format of the Blue Palette Color Lookup table Data (0028,1203). See Section C.7.6.3.1.5 for further explanation.
Palette Color Lookup Table UID	(0028,1199)	3	Palette Color Lookup Table UID. See Section C.7.9.1 for further explanation.
Red Palette Color Lookup Table Data	(0028,1201)	1C	Red Palette Color Lookup Table Data. Required if segmented data is NOT used in an Image IOD or Color Palette IOD, or if the IOD is a Presentation State IOD. See Section C.7.6.3.1.6 for further explanation.
Green Palette Color Lookup Table Data	(0028,1202)	1C	Green Palette Color Lookup Table Data. Required if segmented data is NOT used in an Image IOD or Color Palette IOD, or if the IOD is a Presentation State IOD. See Section C.7.6.3.1.6 for further explanation.
Blue Palette Color Lookup Table Data	(0028,1203)	1C	Blue Palette Color Lookup Table Data. Required if segmented data is NOT used in an Image IOD or Color Palette IOD, or if the IOD is a Presentation State IOD. See Section C.7.6.3.1.6 for further explanation.
Segmented Red Palette Color Lookup Table Data	(0028,1221)	1C	Segmented Red Palette Color Lookup Table Data. Required if segmented data is used in an Image IOD or Color Palette IOD; shall not be present in a Presentation State IOD. See Section C.7.9.2 for further explanation.
Segmented Green Palette Color Lookup Table Data	(0028,1222)	1C	Segmented Green Palette Color Lookup Table Data. Required if segmented data is used in an Image IOD or Color Palette IOD; shall not be present in a Presentation State IOD See Section C.7.9.2 for further explanation.
Segmented Blue Palette Color Lookup Table Data	(0028,1223)	1C	Segmented Blue Palette Color Lookup Table Data. Required if segmented data is used in an Image IOD or Color Palette IOD; shall not be present in a Presentation State IOD. See Section C.7.9.2 for further explanation.

C.7.9.1 Palette Color Lookup Table UID

This Attribute uniquely identifies a palette color lookup table set (red, green, blue).

Note

This can be used to avoid reloading a palette if a system already has that palette loaded without examining all the data entries in the palette.

If this Attribute is present in a Color Palette IOD, it shall have the same value as the SOP Instance UID.

C.7.9.2 Segmented Palette Color Lookup Table Data

The Segmented Palette Color Lookup Table Data (0028,1221-1223) is stored as a series of segments, see Table C.7-23. When the segments are expanded into the actual lookup table data, it shall have the number of table entries specified by the first value of the Palette Color Lookup Table Descriptors (0028,1101-1103), Number of Table Entries.

These lookup tables shall be used only when segmented lookup table data use is desirable and there is a single sample per pixel (single image plane) in the image.

Table C.7-23. Compressed Palette Color Lookup Table Data

Segment 0
Segment 1
…

Segment n

There are currently three types of segments: discrete, linear, and indirect. The segments type is identified by the opcodes in Table C.7-24.

Table C.7-24. Segment Types

Opcode	Segment type
0	Discrete
1	Linear
2	Indirect
3 & above	reserved

C.7.9.2.1 Discrete Segment Type

The discrete segment is used to represent a series of palette components that are not monotonic with respect to their predecessors or successors. The Segment Length indicates the number of lookup table entries.

The format of the Discrete Segment Type shall be as in Table C.7-25.

Table C.7-25. Discrete Segment Type

Segment Opcode = 0
Segment Length
Segment Length number of lookup table entries

C.7.9.2.2 Linear Segment Type

The linear segment represents a series of palette components whose values may be represented by a straight line.

X = palette address, Y = Value contained in the palette.

(X_0, Y_0) = end of the previous segment

(X_0 + Segment Length, Y_1) = end of this linear segment

Where: Y_1 is contained in the data portion of this segment.

During expansion, the application should "connect" the previous segment's endpoint, (X_0, Y_0), with this segment's endpoint, (X_0 + Segment Length, Y_1) using a straight line, by computing the values for each point between the endpoints.

Note

Because the linear segment uses the end point from the previous segment, a linear segment can not be the first segment.

The linear segment's format shall be as in Table C.7-26.

Table C.7-26. Linear Segment Type

Segment Opcode = 1
Segment Length
Y1

C.7.9.2.3 Indirect Segment Type

The indirect segment allows the re-use of repetitive regions within lookup table without respecifying the segment. The opcode is followed by the number of segments to copy and one offset pointer to the first segment to copy. The byte offset is relative to the beginning of

the lookup table. For example, if an indirect segment wants to point to the first segment, then the offset will be zero. The offset is a 32 bit value but is stored in the segment as a least significant 16 bit value followed by a most significant 16 bit value. An indirect segment shall not point to or copy another indirect segment. This avoids the need for recursion and also avoids the possibility of infinite loops.

The indirect segment's format shall be as follows:

Table C.7-27. Indirect Segment Type

Segment Opcode = 2
Number of segments to copy
Least significant 16 bits of byte offset to first segment to copy
Most significant 16 bits of byte offset to first segment to copy

C.7.6.3.1.5 Palette Color Lookup Table Descriptor

The three values of Palette Color Lookup Table Descriptor (0028,1101-1104) describe the format of the Lookup Table Data in the corresponding Data Element (0028,1201-1204) or (0028,1221-1223). In this section, the term "input value" is either the Palette Color Lookup Table input value described in the Enhanced Palette Color Lookup Table Sequence (0028,140B) or if that Attribute is absent, the stored pixel value.

The first Palette Color Lookup Table Descriptor value is the number of entries in the lookup table. When the number of table entries is equal to 2^{16} then this value shall be 0. The first value shall be identical for each of the Red, Green, Blue and Alpha Palette Color Lookup Table Descriptors.

The second Palette Color Lookup Table Descriptor value is the first input value mapped. This input value is mapped to the first entry in the Lookup Table Data. All input values less than the first value mapped are also mapped to the first entry in the Lookup Table Data if the Photometric Interpretation is PALETTE COLOR.

Note

In the case of the Supplemental Palette Color LUT, the stored pixel values less than the second descriptor value are grayscale values.

An input value one greater than the first value mapped is mapped to the second entry in the Lookup Table Data. Subsequent input values are mapped to the subsequent entries in the Lookup Table Data up to an input value equal to number of entries + first value mapped - 1, which is mapped to the last entry in the Lookup Table Data. Input values greater than or equal to number of entries + first value mapped are also mapped to the last entry in the Lookup Table Data. The second value shall be identical for each of the Red, Green, Blue and Alpha Palette Color Lookup Table Descriptors.

The third Palette Color Lookup Table Descriptor value specifies the number of bits for each entry in the Lookup Table Data. It shall take the value of 8 or 16. The LUT Data shall be stored in a format equivalent to 8 bits allocated when the number of bits for each entry is 8, and 16 bits allocated when the number of bits for each entry is 16, where in both cases the high bit is equal to bits allocated-1. The third value shall be identical for each of the Red, Green and Blue Palette Color Lookup Table Descriptors.

Note

Some implementations have encoded 8 bit entries with 16 bits allocated, padding the high bits; this can be detected by comparing the number of entries specified in the LUT Descriptor with the actual value length of the LUT Data entry. The value length in bytes should equal the number of entries if bits allocated is 8, and be twice as long if bits allocated is 16.

When the Red, Green, or Blue Palette Color Lookup Table Descriptor (0028,1101-1103) are used as part of the Palette Color Lookup Table Module or the ??? in an Image **(other than a Segmentation)** or Presentation State IOD, the third value shall be equal to 16. When the Alpha Palette Color Lookup Table Descriptor (0028,1104) is used, the third value shall be equal to 8.

When the Red, Green, or Blue Palette Color Lookup Table Descriptor (0028,1101-1103) are used as part of the Palette Color Lookup Table Module in a Color Palette IOD, the 3rd value of Palette Color Lookup Table Descriptor (0028,1101-1103) (i.e, the number of bits for each entry in the Lookup Table Data) shall be 8.

When the Red, Green, or Blue Palette Color Lookup Table Descriptor (0028,1101-1103) are used as part of the Palette Color Lookup Table Module in a Segmentation IOD, the 3rd value of Palette Color Lookup Table Descriptor (0028,1101-1103) (i.e, the number of bits for each entry in the Lookup Table Data) shall be 8 or 16.

Note

1. A value of 16 indicates the Lookup Table Data will range from (0,0,0) minimum intensity to (65535,65535,65535) maximum intensity.

2. Since the Palette Color Lookup Table Descriptor (0028,1101-1104) Attributes are multi-valued, in an Explicit VR Transfer Syntax, only one value representation (US or SS) may be specified, even though the first and third values are always by definition interpreted as unsigned. The explicit VR actually used is dictated by the VR needed to represent the second value, which will be consistent with Pixel Representation (0028,0103).

C.7.6.3.1.6 Palette Color Lookup Table Data

Palette Color Lookup Table Data (0028,1201-1204) contain the lookup table data corresponding to the Lookup Table Descriptor (0028,1101-1104).

Palette color values must always be scaled across the full range of available intensities. This is indicated by the fact that there are no bits stored and high bit values for palette color data.

Note

For example, if there are 16 bits per entry specified and only 8 bits of value are truly used then the 8 bit intensities from 0 to 255 must be scaled to the corresponding 16 bit intensities from 0 to 65535. To do this for 8 bit values, simply replicate the value in both the most and least significant bytes.

These lookup tables shall be used only when there is a single sample per pixel (single image plane) in the image.

PS3.4 DICOM PS3.4 - Service Class Specifications

*Amend PS3.4 as follows (changes to existing text are bold and **underlined** for additions and **~~struckthrough~~** for removals):*

B.5 Standard SOP Classes

The SOP Classes in the Storage Service Class identify the Composite IODs to be stored. Table B.5-1 identifies Standard SOP Classes.

Table B.5-1. Standard SOP Classes

SOP Class Name	SOP Class UID	IOD Specification (defined in PS3.3)	Specialization
...			
Digital X-Ray Image Storage - For Presentation	1.2.840.10008.5.1.4.1.1.1.1	Digital X-Ray Image IOD	B.5.1.1
Digital X-Ray Image Storage - For Processing	1.2.840.10008.5.1.4.1.1.1.1.1	Digital X-Ray Image IOD	B.5.1.1
...			
Segmentation Storage	1.2.840.10008.5.1.4.1.1.66.4	Segmentation IOD	**B.5.1.n1**
Label Map Segmentation Storage	**1.2.840.10008.5.1.4.1.1.66.nn**	Segmentation IOD	**B.5.1.n1**
Surface Segmentation Storage	1.2.840.10008.5.1.4.1.1.66.5	Surface Segmentation IOD	
...			

B.5.1 Specialization for Standard SOP Classes

B.5.1.1 Digital X-Ray Image Storage SOP Classes

The Digital X-Ray Image Storage - For Presentation SOP Class shall use the DX IOD with an Enumerated Value of FOR PRESENTATION for Presentation Intent Type (0008,0068).

The Digital X-Ray Image Storage - For Processing SOP Class shall use the DX IOD with an Enumerated Value of FOR PROCESSING for Presentation Intent Type (0008,0068).

...

B.5.1.n1 Segmentation Storage SOP Classes

The Segmentation Storage SOP Class shall use the Segmentation IOD with an Enumerated Value of BINARY or FRACTIONAL for Segmentation Type (0062,0001).

The Label Map Segmentation Storage SOP Class shall use the Segmentation IOD with an Enumerated Value of LABELMAP for Segmentation Type (0062,0001).

PS3.6 DICOM PS3.6 - Data Dictionary

*Amend PS3.6 as follows (changes to existing text are bold and **underlined** for additions and **~~struckthrough~~** for removals):*

A Registry of DICOM Unique Identifiers (UIDs) (Normative)

Table A-1. UID Values

UID Value	UID Name	UID Keyword	UID Type	Part
1.2.840.10008.5.1.4.1.1.66.nn	**Label Map Segmentation Storage**	**LabelMapSegmentation Storage**	**SOP Class**	**PS3.4**