

2024/11/14

The following changes have been made relative to the previously published PS3 2024d release of the standard, by incorporating the changes specified in the supplements and correction items.

The Final Text of all applied Supplements and Correction Proposals is available at <https://medical.nema.org/medical/dicom/final/>

Production Notes

The DocBook XML files are the source format, and all other formats are rendered from it.

The PDF format is rendered from the DocBook XML, and remains the "official" (authoritative) form of the standard. The PDF contains hyperlinks to sections, figures and tables both within and between parts (which in the latter case work if you are reading the PDF in a tool that supports linking to other parts).

The two HTML formats are provided for the convenience of those who find them easier to navigate within a browser, and though the appearance and organization is different, the content is the same. One form consists of entire parts in one very large HTML page, and the other consist of chunks of sections with navigation elements. Both forms are hyper-linked within and between parts. The figures in the HTML are SVG, so a browser that supports SVG is required (most contemporary browsers do).

All paragraphs (<p/> elements) in the HTML files of this release, are uniquely identified with a hypertext anchor (<a/> element), each of which has an id attribute (derived from the source DocBook <para/> element xml:id attribute). These unique identifiers will remain stable in subsequent releases, so they may be reliably used as the persistent targets of hyperlinks relative to the current release base URL, and are more specific than the existing anchors for entire sections or tables. Unlike the section and table anchors, there is no semantic significance to the syntax of the identifiers (i.e., they are UUIDs, rather than being derived from the section or table numbering pattern). Subsequent releases will add new identifiers for new paragraphs and text split out of existing paragraphs into new paragraphs, and will, if possible, empty, rather than entirely remove, existing paragraphs that are retired (in order to avoid dead links).

The chunked HTML format includes navigation elements in the header and footer, as well as a hyperlink to the current release of that page, in case the user happens to find or be using an older release of the page.

The DOCX (for Word) and ODT (for OpenOffice or LibreOffice) formats are provided for the convenience of future Supplement and CP editors. Their main claim to fame is that they exist at all, and though they are viewable and editable, they are lacking many features of the Word source of previous release, for example the use of styles for section headings. They do contain embedded hyperlinks, and these are also present in the table of contents, even though the page numbers rendered in the table of contents may be meaningless. To reiterate, the intent of these files is to provide a source to cut and past into new Word documents, and not to be functional documents in their own right. Since Word does not support SVG, all figures embedded in the DOCX files have been rasterized to a fixed resolution and are adequate for position only and are not editable and are not intended to be a substitute for the SVG figures.

The rendering pipeline used to produce these files is available but requires some expertise to use it. It is not supported. To achieve quality rendering, the use of some commercial tools was necessary, to supplement the many open source tools that were also used. Oxygen (commercial) was used as the XML editor since it supports a WYSIWG authoring mode. OpenOffice (open source) was used as the equation editor. The DocBook (open source, version docbook-xsl-ns-1.78.1) style sheets were used to create the HTML and intermediate FO form used to create the PDF and DOCX. MathML equations were converted to SVG using pMML2SVG (open source, version pMML2SVG-0.8.5). RenderX XEP (commercial) was used to produce the PDF, and XMLmind FO-Converter (commercial) was used to produce the DOCX. The difference files were produced using DeltaXML DocBook Compare (commercial). The PDF files were post-processed with qpdf to generate object streams to reduce the size of the tagged PDF and improve searching for strings that span lines within tables and to linearize the files for streamed web page viewing.

Some characteristics of the DocBook XML may be of interest to those performing automated processing or extraction:

- Zero width spaces (U+200B) are used in some places to allow long words (such as PS3.6 keywords and UIDs) to break within table columns and avoid tables becoming too wide to fit on a page. These need to be filtered out before using these words literally.
- Enumerated values and defined terms are formalized in PS3.3 as DocBook variablelist elements with a title identifying them as such, to facilitate their automated detection and extraction.
- Template and context group tables in PS 3.16 are preceded by variablelist elements defining whether or not they are extensible, etc., again to enable automated extraction.

- Hyperlinks (xref and link elements) are used extensively but may obscure the identifier of what is being linked to from the perspective of automated extraction. It may be useful to consult the olink targetdb files that are included in the package to "look up" the target of such links, rather than reinventing this mechanism, which is used by the DocBook stylesheets for cross-document linking. E.g., one can look up "sect_TID_300" in "output/html/targetdb/PS3_16_target.db" to determine that it has a "number" of "TID 300" and a "ttl" of "Measurement", etc.

Changes to Parts

General Changes

PS3.1

PS3.2

- CP 2405

PS3.3

- Move inclusion of Table C.36.2.4.5-2 to CT Image Module, not RT Image Module (CP 2220)
- Clarify use of release vs. version of standard
- For Instance Origin Status, factored out excessively long text into sub-section and referenced it
- Use full data element name and tag for SOP Instance UID in Derivation Description
- CP 2342
- CP 2346
- CP 2370
- CP 2388
- CP 2389
- CP 2400
- CP 2401
- CP 2402
- CP 2404

PS3.4

- Correct Destination AE tag in Table CC.2.5-2g
- For requirement column headers, always use parentheses and change all Requirement Type to Usage for consistency

PS3.5

- Qualify conditional data element definition with 'unless it is specified that they may be present otherwise'
- CP 2388
- CP 2389
- CP 2396

PS3.6

- CP 2346
- CP 2401
- CP 2402

PS3.7

- CP 2405

PS3.8**PS3.10****PS3.11****PS3.12****PS3.14****PS3.15**

- CP 2399

PS3.16

- CP 2319
- CP 2393
- CP 2398
- CP 2401

PS3.17**PS3.18**

- Capitalize Header Field, Media Type a la Query Parameter throughout
- CP 2389
- CP 2396

PS3.19

- CP 2396

PS3.20

PS3.21

PS3.22

Supplements Incorporated

Correction Items Incorporated

- CP 2319** Clarify conditions in TID 10007
- CP 2342** Add Clinical Fraction Number to 1st Generation RT Objects
- CP 2346** Ocular Region Imaged Module in Ophthalmic Tomography
- CP 2370** Orientation guidance for vertical CT gantries
- CP 2388** Extended Offset Table clarification
- CP 2389** Define Fragment
- CP 2393** Update CT Scanning Length Definition
- CP 2396** JIS X 201 US-ASCII differences affecting delimiter and private creators
- CP 2398** Correct inconsistent time point concepts
- CP 2399** Correct Audit Message Schema Conflict
- CP 2400** Add Mixed Ion Radiation Type and add Ion species information in control points
- CP 2401** Ion Range Surface Compensator
- CP 2402** Isodose levels for RT Dose visualization
- CP 2404** Rephrase Reference Description
- CP 2405** Add note clarifying DIMSE JWT and DICOMweb access token can be the same