

| | | |
|----|---|----------------------------|
| 1 | Status | Letter Ballot |
| 2 | Date of Last Update | 2025/06/17 |
| 3 | Person Assigned | David Clunie |
| 4 | | mailto:dclunie@dclunie.com |
| 5 | Submitter Name | David Clunie |
| 6 | | mailto:dclunie@dclunie.com |
| 7 | Submission Date | 2024/11/13 |
| 8 | Correction Number CP-2480 | |
| 9 | Log Summary: Clarify application/octet-stream can only contained uncompressed data | |
| 10 | Name of Standard | |
| 11 | PS3.18 | |
| 12 | Rationale for Correction: | |
| 13 | Media types for uncompressed and compressed images are clearly distinguished in PS3.18, but because there is an optional | |
| 14 | parameter that allows the transfer syntax to be specified, some implementations have mistakenly responded with a compressed | |
| 15 | octet stream in application/octet-stream. | |
| 16 | Clarify that this is not permitted, and that the appropriate transfer syntax specific media type should be used in the request, which | |
| 17 | for compressed images may be an image/* wildcard if necessary, or */* if it is unknown whether the image is compressed or not. | |
| 18 | Also, when a Transfer Syntax for Encapsulated Uncompressed Explicit VR Little Endian was added in CP 2083, there was no | |
| 19 | consideration of the appropriate media type to use. Clarify this. | |
| 20 | <i>[Ed.Note: There is no media type associated with Deflated Explicit VR Little Endian Transfer Syntax (entire dataset deflated, as</i> | |
| 21 | <i>distinct from individual frames); addressing this, if required, could be the subject of a future CP. This could affect the "no defined</i> | |
| 22 | <i>compression transfer syntaxes for Text/Other" entries, for example.]</i> | |
| 23 | Correction Wording: | |

Amend DICOM PS3.18 as follows (changes to existing text are bold and underlined for additions and ~~struckthrough~~ for removals):

8.7.3.3 Bulkdata Media Types

Bulkdata representations are only supported by RESTful services. There are two categories of Bulkdata: uncompressed and compressed.

The Selected Media Type will be the default Media Type for the Resource Category when the origin server supports none of the Acceptable Media Types, as described in ???, unless the origin server has only access to the pixel data in lossy compressed form or the pixel data in a lossless compressed or encapsulated uncompressed form that is of such length that it cannot be encoded in the Explicit VR Little Endian Transfer Syntax.

The origin server may support additional Transfer Syntaxes.

If no Media Type Transfer Syntax parameter is specified, then the Explicit VR Little Endian Transfer Syntax "1.2.840.10008.1.2.1" shall be used, unless the origin server has only access to the pixel data in lossy compressed form or the pixel data in a lossless compressed or encapsulated uncompressed form that is of such length that it cannot be encoded in the Explicit VR Little Endian Transfer Syntax.

Note

The tables in this section have no entries for the URI service, since they do not support separate retrieval of Bulkdata.

Depending on the Selected Media Type, the pixel data of a resource in the Single Frame Image Resource Category is encoded in:

- one compressed Bulkdata representation, or
- one uncompressed Bulkdata representation.

Depending on the Selected Media Type, the pixel data of a resource in the Multi-Frame Image Resource Category is encoded in:

- multiple Single Frame Image compressed Bulkdata representations: one for each frame, or
- one Multi-Frame Image uncompressed Bulkdata representation.

Depending on the Selected Media Type, the pixel data of a resource in the Video Resource Category is encoded in:

- one Video compressed Bulkdata representation, or
- one Video uncompressed Bulkdata representation.

8.7.3.3.1 Uncompressed Bulkdata Media Types

Table 8.7.3-4 specifies the default Media Type and Transfer Syntax UIDs, by Resource Category (see ???) that can be used with uncompressed Bulkdata for the RESTful services. Uncompressed Bulkdata is encoded as a stream of uncompressed bytes (octets) in Little Endian byte order.

Note

1. This is the same encoding defined in PS3.19 for the returned value of the getData() call for uncompressed Bulkdata.
2. In a Multi-Frame Image with a Bits Allocated (0028,0100) of 1 that is uncompressed, the individual frames are not padded, therefore successive bits are packed into bytes or words in Native format as described in Section 8.2 "Native or Encapsulated Format Encoding" in PS3.5. This means that if only selected frames of a Multi-Frame Image are to be encoded in the response, each frame needs to be extracted from the Multi-Frame Image pixel data and successively concatenated in the response, with no padding at the start of first byte in the response, and with no padding between successive encoded frames in the response. I.e., all the frame-specific bitstreams are successively encoded with no padding at the beginning or in between.
3. Only uncompressed Bulkdata is permitted to be returned with a Media Type as application/octet-stream. Specific media types other than application/octet-stream are defined for each kind of compressed Bulkdata in Section 8.7.3.3.2. This means that if a wildcard Transfer Syntax parameter ("transfer-syntax=") is included with the Media Type, that it is only applicable to uncompressed Transfer Syntaxes.

Table 8.7.3-4. Transfer Syntax UUIDs for Uncompressed Data in Bulkdata

| Category | Media Type | Transfer Syntax UUID | Transfer Syntax Name | RESTful |
|--------------------|--------------------------|--------------------------------------|---|-----------------|
| Single Frame Image | application/octet-stream | 1.2.840.10008.1.2.1 | Explicit VR Little Endian | D |
| | | <u>1.2.840.10008.1.2.1.98</u> | <u>Encapsulated Uncompressed Explicit VR Little Endian</u> | <u>O</u> |
| Multi-Frame Image | application/octet-stream | 1.2.840.10008.1.2.1 | Explicit VR Little Endian | D |
| | | <u>1.2.840.10008.1.2.1.98</u> | <u>Encapsulated Uncompressed Explicit VR Little Endian</u> | <u>O</u> |
| Video | application/octet-stream | 1.2.840.10008.1.2.1 | Explicit VR Little Endian | D |
| | | <u>1.2.840.10008.1.2.1.98</u> | <u>Encapsulated Uncompressed Explicit VR Little Endian</u> | <u>O</u> |
| Text | application/octet-stream | 1.2.840.10008.1.2.1 | Explicit VR Little Endian | D |
| Other | application/octet-stream | 1.2.840.10008.1.2.1 | Explicit VR Little Endian | D |

Note

Even though the Transfer Syntax is Explicit VR Little Endian, the Value Representation is not actually encoded at the beginning of the octet-stream. The Value Representation is contained in the Metadata that references the Bulkdata. **For this reason, the Implicit VR Little Endian Transfer Syntax is not listed in this table.**

If Encapsulated Uncompressed Explicit VR Little Endian is supported, the response includes the octet stream without the fragment delimiters. I.e., the octet stream is unencapsulated before it is transferred.

8.7.3.3.2 Compressed Bulkdata Media Types

Compressed Bulkdata contains only the compressed octet stream without the fragment delimiters.

Table 8.7.3-5 specifies the default and optional Media Types and Transfer Syntax UID combinations for each Resource Category (see ???) of compressed Bulkdata for the RESTful services.

Note

1. Some of the Transfer Syntax Names include text about Default Transfer Syntax, however this applies to its role in DIMSE transactions, rather than the default for RESTful services (which is specified in the RESTful column of the table).
2. The Media Type column reflects the data encoding but does not include extended Media Type descriptors such as "multipart/related" that describe further packaging of the encoded data.

These Media Types can be used to retrieve Bulkdata, such as images or video, encoded in a specific Transfer Syntax.

Note

1. **These Media Types are not used for uncompressed Bulkdata, for which the use of application/octet-stream is specified in Section 8.7.3.3.1. Compressed Bulkdata is never transferred using application/octet-stream.**
2. **If a wildcard Transfer Syntax parameter ("transfer-syntax=**") is included with the Media Type, that it is only applicable to compressed Transfer Syntaxes corresponding to the specified Media Type.**
3. **If an image-specific wildcard Media Type is requested with "image/**", then the server is free to decided what compressed encoding it wants to return.**

For details on how Compressed Bulkdata is packaged into single part or multipart payloads, see ???.

Table 8.7.3-5. Media Types and Transfer Syntax UUIDs for Compressed Data in Bulkdata

| Resource Category | Media Type | Transfer Syntax UUID | Transfer Syntax Name | Optionality |
|--------------------|-----------------------|-------------------------|--|-------------|
| Single Frame Image | image/jpeg | 1.2.840.10008.1.2.4.70 | JPEG Lossless, Non-Hierarchical, First-Order Prediction(Process 14 [Selection Value 1]): Default Transfer Syntax for Lossless JPEG Image Compression | D |
| | | 1.2.840.10008.1.2.4.50 | JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression | O |
| | | 1.2.840.10008.1.2.4.51 | JPEG Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only) | O |
| | | 1.2.840.10008.1.2.4.57 | JPEG Lossless, Non-Hierarchical (Process 14) | O |
| | application/x-deflate | 1.2.840.10008.1.2.8.1 | Deflated Image Frame Compression | D |
| | image/dicom-rle | 1.2.840.10008.1.2.5 | RLE Lossless | D |
| | image/jls | 1.2.840.10008.1.2.4.80 | JPEG-LS Lossless Image Compression | D |
| | | 1.2.840.10008.1.2.4.81 | JPEG-LS Lossy (Near-Lossless) Image Compression | O |
| | image/jp2 | 1.2.840.10008.1.2.4.90 | JPEG 2000 Image Compression (Lossless Only) | D |
| | | 1.2.840.10008.1.2.4.91 | JPEG 2000 Image Compression | O |
| | image/jpx | 1.2.840.10008.1.2.4.92 | JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only) | D |
| | | 1.2.840.10008.1.2.4.93 | JPEG 2000 Part 2 Multi-component Image Compression | O |
| | image/jphc | 1.2.840.10008.1.2.4.201 | High-Throughput JPEG 2000 Image Compression (Lossless Only) | D |
| | | 1.2.840.10008.1.2.4.202 | High-Throughput JPEG 2000 with RPCL Options Image Compression (Lossless Only) | O |
| | | 1.2.840.10008.1.2.4.203 | High-Throughput JPEG 2000 Image Compression | O |
| | image/jxl | 1.2.840.10008.1.2.4.110 | JPEG XL Lossless | D |
| | | 1.2.840.10008.1.2.4.111 | JPEG XL JPEG Recompression | O |
| | | 1.2.840.10008.1.2.4.112 | JPEG XL | O |
| Multi-frame Image | image/jpeg | 1.2.840.10008.1.2.4.70 | JPEG Lossless, Non-Hierarchical, First-Order Prediction(Process 14 [Selection Value 1]): Default Transfer Syntax for Lossless JPEG Image Compression | D |
| | | 1.2.840.10008.1.2.4.50 | JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression | O |
| | | 1.2.840.10008.1.2.4.51 | JPEG Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only) | O |
| | | 1.2.840.10008.1.2.4.57 | JPEG Lossless, Non-Hierarchical (Process 14) | O |
| | application/x-deflate | 1.2.840.10008.1.2.8.1 | Deflated Image Frame Compression | D |
| | image/dicom-rle | 1.2.840.10008.1.2.5 | RLE Lossless | D |
| | image/jls | 1.2.840.10008.1.2.4.80 | JPEG-LS Lossless Image Compression | D |
| | | 1.2.840.10008.1.2.4.81 | JPEG-LS Lossy (Near-Lossless) Image Compression | O |
| | image/jp2 | 1.2.840.10008.1.2.4.90 | JPEG 2000 Image Compression (Lossless Only) | D |

| Resource Category | Media Type | Transfer Syntax UID | Transfer Syntax Name | Optionality |
|-------------------|-----------------------|--|---|-------------|
| | image/jpx | 1.2.840.10008.1.2.4.91 | JPEG 2000 Image Compression | O |
| | | 1.2.840.10008.1.2.4.92 | JPEG 2000 Part 2 Multi-component Image Compression (Lossless Only) | D |
| | | 1.2.840.10008.1.2.4.93 | JPEG 2000 Part 2 Multi-component Image Compression | O |
| | image/jphc | 1.2.840.10008.1.2.4.201 | High-Throughput JPEG 2000 Image Compression (Lossless Only) | D |
| | | 1.2.840.10008.1.2.4.202 | High-Throughput JPEG 2000 with RPCL Options Image Compression (Lossless Only) | O |
| | | 1.2.840.10008.1.2.4.203 | High-Throughput JPEG 2000 Image Compression | O |
| | image/jxl | 1.2.840.10008.1.2.4.110 | JPEG XL Lossless | D |
| | | 1.2.840.10008.1.2.4.111 | JPEG XL JPEG Recompression | O |
| | | 1.2.840.10008.1.2.4.112 | JPEG XL | O |
| Video | video/mpeg | 1.2.840.10008.1.2.4.100 | MPEG2 Main Profile @ Main Level | O |
| | | 1.2.840.10008.1.2.4.100.1 | Fragmentable MPEG2 Main Profile @ Main Level | O |
| | | 1.2.840.10008.1.2.4.101 | MPEG2 Main Profile @ High Level | D |
| | | 1.2.840.10008.1.2.4.101.1 | Fragmentable MPEG2 Main Profile @ High Level | O |
| | video/mp4 | 1.2.840.10008.1.2.4.102 | MPEG-4 AVC/H.264 High Profile / Level 4.1 | D |
| | | 1.2.840.10008.1.2.4.102.1 | Fragmentable MPEG-4 AVC/H.264 High Profile / Level 4.1 | D |
| | | 1.2.840.10008.1.2.4.103 | MPEG-4 AVC/H.264 BD-compatible High Profile / Level 4.1 | O |
| | | 1.2.840.10008.1.2.4.103.1 | Fragmentable MPEG-4 AVC/H.264 BD-compatible High Profile / Level 4.1 | O |
| | | 1.2.840.10008.1.2.4.104 | MPEG-4 AVC/H.264 High Profile / Level 4.2 For 2D Video | O |
| | | 1.2.840.10008.1.2.4.104.1 | Fragmentable MPEG-4 AVC/H.264 High Profile / Level 4.2 For 2D Video | O |
| | | 1.2.840.10008.1.2.4.105 | MPEG-4 AVC/H.264 High Profile / Level 4.2 For 3D Video | O |
| | | 1.2.840.10008.1.2.4.105.1 | Fragmentable MPEG-4 AVC/H.264 High Profile / Level 4.2 For 3D Video | O |
| | | 1.2.840.10008.1.2.4.106 | MPEG-4 AVC/H.264 Stereo High Profile / Level 4.2 | O |
| | | 1.2.840.10008.1.2.4.106.1 | Fragmentable MPEG-4 AVC/H.264 Stereo High Profile / Level 4.2 | O |
| | video/H265 | 1.2.840.10008.1.2.4.107 | HEVC/H.265 Main Profile / Level 5.1 | D |
| | | 1.2.840.10008.1.2.4.108 | HEVC/H.265 Main 10 Profile / Level 5.1 | O |
| | application/x-deflate | 1.2.840.10008.1.2.8.1 | Deflated Image Frame Compression | D |
| Text | | N/A (no defined compression transfer syntaxes for Text) | | |
| Other | | N/A (no defined compression transfer syntaxes for Other) | | |

The origin server may support additional Transfer Syntaxes.

For the Media Type image/jpeg Transfer Syntaxes, the image may or may not include the JFIF marker segment. The image may or may not include APP2 marker segments with an identifier of "ICC_PROFILE". There is no requirement for the origin server to add a

1 JFIF marker segment nor to copy the value of the ICC Profile (0028,2000) Attribute, if present, into APP2 marker segments in the
2 compressed data stream. See Section 8.2.1 "JPEG Image Compression" in PS3.5.

3 For the Media Type image/jp2 and image/jpx Transfer Syntaxes, the image does not include the jp2 marker segment. See Section
4 8.2.4 "JPEG 2000 Image Compression" in PS3.5 and Section A.4.4 "JPEG 2000 Image Compression" in PS3.5

5 Compressed multi-frame image pixel data is encoded as individual frames. E.g., each frame of a JPEG 2000 multi-frame image will
6 be encoded separately as image/jp2 representations, rather than as a single video/mj2 (???) or application/octet-stream representation.
7 See ??? for details on how multiple representations can be packaged into a multipart payload.

8 Video pixel data is encoded as a single video representation. E.g., all frames of an MPEG-4 video will be encoded as a single video/mp4
9 (???) representation.

10 **Note**

11 1. The resource on the origin server may have been encoded in the Deflated Explicit VR Little Endian
12 (1.2.840.10008.1.2.1.99) Transfer Syntax. If so, the origin server may inflate it, and then convert it into an Acceptable
13 Transfer Syntax. A Content-Encoding header field of 'deflate' cannot be used to transfer the deflated bytes unaltered
14 since the required PS3.10 File Meta Information is not included in the deflated bytes, and the Content-Encoding applies
15 to the entire multipart stream, not each part within it individually.

16 The resource on the origin server may have been encoded in the Deflated Image Frame Compression
17 (1.2.840.10008.1.2.8.1) Transfer Syntax. If so, the origin server may return the compressed bit stream if it is an Acceptable
18 Transfer Syntax, or the origin server may inflate it, and then convert it into an Acceptable Transfer Syntax. Alternatively,
19 if the user agent allowed a Content-Encoding header field of 'deflate', then the deflated bytes for a single part response
20 may be transferred after adding a zlib container per ??? Section 4.2.2, but the Transfer Syntax parameter in the response
21 should be the Explicit VR Little Endian Transfer Syntax.

22 2. Many of the Media Types used for compressed Pixel Data transferred as Bulkdata values are also used for consumer
23 format Media Types. A web browser may not be able to display the encoded data directly, even though some of the
24 same Media Types are also used for encoding rendered Pixel Data. See ???.

25 For example, the Media Type for Bulkdata values of lossless 16-bit JPEG ??? encoded Pixel Data is "image/jpeg", the
26 same Media Type as might be used for 8-bit JPEG ??? encoded Pixel Data, whether extracted as Bulkdata, or rendered.
27 The Transfer Syntax parameter of the Content-Type Header Field is useful to signal the difference.

28 3. Previously, experimental Media Types "image/x-dicom-rle" and "image/x-jls" were defined, so origin servers and user
29 agents may want to account for these when communicating with older implementations. These have been replaced with
30 the standard Media Types "image/dicom-rle" and "image/jls", respectively.

31 **8.7.3.5 Media Type Syntax**

32 The syntax of Media Type usage in DICOM is:

33 dicom-media-type = (dcm-singlepart / dcm-multipart) [dcm-parameters]

34 Where

35 dcm-singlepart = dcm-mt-name

36 dcm-multipart ;see Section 8.7.3.5.1

37 dcm-parameters = transfer-syntax-mtp ;see Section 8.7.3.5.2

38 / charset-mtp;see ???

39 dcm-mt-name = dicom / dicom-metadata / bulkdata / pixeldata ;DICOM Media Type name

40 dicom = "application/dicom"

41 dicom-metadata = dicom-xml / dicom-json

1 dicom-xml = "application/dicom+xml"

2 dicom-json = "application/dicom+json"

3 bulkdata = octet-stream / pixeldata

4 octet-stream = "application/octet-stream"

5 pixeldata = image-pixel / video-pixel

6 rendered = image-pixel / video-pixel

7 image-pixel = "image/jpeg" / "image/dicom-rle" / "image/jls" / "image/jp2" / "image/jpx" / "image/jphc" / "image/jxl"

8 video-pixel = "video/mpeg" / "video/mp4" / "video/H265"

9 All Media Types used in DICOM may have a Transfer Syntax parameter, but its usage may be constrained by the service for which
10 they are used.

11 **Note**

12 The application/dicom+xml and application/dicom+json Media Types may have a Transfer Syntax parameter in order to
13 specify the encoding of base64 data.

14 All Media Types used in DICOM may have a character set parameter, but its usage may be constrained by the service for which they
15 are used.

16 **8.7.3.5.1 Multipart Media Types**

17 The syntax of multipart Media Types is:

18 dcm-multipart = "multipart/related"

19 OWS ";" OWS "type" "=" dcm-mp-mt-name

20 OWS ";" OWS "boundary=" boundary

21 [dcm-parameters]

22 [related-parameters]

23 Where

24 dcm-mp-mt-name = dicom-mt-name / rendered

25 See ??? for the definition of boundary and related-parameters.

26 Each multipart Media Type shall include a "type" parameter that defines the Media Type of the parts and shall also include a
27 "boundary" parameter that specifies the boundary string that is used to separate the parts. For example:

28 Accept: multipart/related; type="application/octet-stream", multipart/related; type="image/*"; boundary=**, multipart/related

29 **8.7.3.5.2 Transfer Syntax Parameter**

30 For a given DICOM Media Type, a single Transfer Syntax parameter value may be specified, but its usage may be constrained by
31 the service for which they are used.

32 RESTful origin servers shall support the Transfer Syntax parameter.

33 Transfer syntax Media Type parameters are forbidden in URI Service requests and responses.

34 The syntax is:

35 transfer-syntax-mtp = OWS ";" OWS %s"transfer-syntax=" ts-value

1 ts-value = transfer-syntax-uid / ""
2 transfer-syntax-uid ; a UID from Table A-1 "UID Values" in PS3.6 with a UID Type of Transfer Syntax
3 The value of the Transfer Syntax parameter may be either a Transfer Syntax UID or the token "".

4 For example, to specify that 1.2.840.10008.1.2.4.50 is the acceptable Transfer Syntaxes, an Accept Header Field could be:
5 Accept: application/dicom; transfer-syntax=1.2.840.10008.1.2.4.50
6 A DICOM Media Type may only have one Transfer Syntax parameter and it shall have only one value.

7 **Note**

8 Per ??? Media Type Specifications and Registration Procedures, it is an error for a specific parameter to be specified more
9 than once. If a choice of Transfer Syntaxes is acceptable. more than one Media Type may be provided in the Accept header
10 with different q parameter values to indicate preference. E.g., to specify that 1.2.840.10008.1.2.4.50 and to specify that
11 1.2.840.10008.1.2.4.57 are acceptable but 1.2.840.10008.1.2.4.50 is preferred, an Accept Header Field could be:

12 Accept: multipart/related; type="application/dicom";transfer-syntax=1.2.840.10008.1.2.4.50;boundary=**, multipart/related

13 The wildcard value "" indicates that the user agent will accept any Transfer Syntax. This allows, for example, the origin server to respond
14 without needing to transcode an existing representation to a new Transfer Syntax, or to respond with the Explicit VR Little Endian
15 Transfer Syntax regardless of the Transfer Syntax stored, unless the origin server has only access to the pixel data in lossy compressed
16 form or the pixel data in a lossless compressed form that is of such length that it cannot be encoded in the Explicit VR Little Endian
17 Transfer Syntax.

18 **Note**

19 The wild card value is applicable only to those Transfer Syntaxes appropriate to the specified Media Type. E.g., only
20 uncompressed Transfer Syntaxes are appropriate for application/octet-stream, and only the relevant compressed
21 Transfer Syntaxes are appropriate for image/* or more specific sub-types of the image Media Type.

22 If an Origin server supports the Transfer Syntax parameter, it shall support the wildcard value.

23 Origin servers that support the Transfer Syntax parameter shall specify in their Conformance Statement those values of Transfer
24 Syntax parameter that are supported in the response.

25 User agents that support the Transfer Syntax parameter shall specify in their Conformance Statement those Transfer Syntax para-
26 meter values that may be supplied in the request.