

DICOM Change Proposal

STATUS	Letter Ballot
Date of Last Update	2025-09-07
Person Assigned	David Clunie <dclunie@dclunie.com>
Submitter Name	Jörg Riesmeier <dicom@jriesmeier.com>
Submission Date	2025-02-06

Change Number	CP-2527
Log Summary: Consistently use defined terms "Bulk Data" and "Bulk Data URI" in PS3.18	
Name of Standard PS3.2, PS3.18	
<p>Rationale for Change:</p> <p>DICOM PS3.18 Section 3.19 defines "Bulk Data" and "Bulk Data URI", but other sections of this Part use "Bulkdata", or "bulk data" if the general concept of a huge amount of data is meant, and "Bulkdata URI". Even "Bulk Data URI" is defined as "A Uniform Resource Identifier that references Bulkdata.", i.e., it uses "Bulkdata" instead of the previously defined "Bulk Data".</p> <p>It is proposed to fix these inconsistencies and use clearly defined terms where appropriate.</p> <p><i>Editorial Instructions:</i></p> <ul style="list-style-type: none">• Change PS3.2 and PS3.18 throughout to use "Bulk Data" rather than "Bulkdata" or "bulk data" in every occurrence other than when used in the syntax of a resource or query parameter, or when referring to the PS3.19 Infoset element BulkData.• Capitalize "DICOM Object" throughout PS3.2 and PS3.18.	
Change Wording:	

5 **For reference PS3.2 Section 3.9 (unchanged)**

3.13 Web Services Definitions

This Part makes use of the following terms defined in PS3.18

Bulk Data See Bulk Data in PS3.18.

Bulk Data URI See Bulk Data URI in PS3.18.

[...]

Modify PS3.18 Section 3.9 as indicated

(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):

3.9 Web Services Definitions

Bulk Data An object that contains an octet-stream containing one or more Value Fields (typically containing large data, such as Pixel Data) extracted from a DICOM Dataset. See Metadata.

Note

1. The octet-stream does not include the Attribute Tag, Value Representation, or Attribute Length.
2. For the value of a Frame of a Pixel Data Attribute encoded in an Encapsulated Transfer Syntax, it does not include the Basic Offset Table and Pixel Data Stream Fragment Item Tags and lengths.

Bulk Data URI A Uniform Resource Identifier that references BulkdData.

DICOM Object An instance of a data object as defined by PS3.3 that has been allocated an unique identifier in the format specified for SOP Instance UID in PS3.3 and has been chosen as an object to be saved securely for some period of time. Within the DICOM Standard, a DICOM Object is typically a Composite Service Object Pair (SOP) Instance.

DICOM Resource One or more DICOM Objects that are referenced by a URL.

[...]

Metadata A DICOM Dataset where zero or more elements (typically containing large data, such as Pixel Data) have been replaced with BulkdData URIs.

Note

Metadata does not include the Group 0002 File Meta Information Data Elements, which describe but are not part of a Dataset, per Section 7.1 in PS3.10.

Modify PS3.18 Chapter 6 as indicated

(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):

6 Conformance

An implementation claiming conformance to this Part of the Standard shall function in accordance with all its mandatory sections.

45 DICOM Web Services are used to transmit Composite SOP Instances. All Composite SOP Instances transmitted shall conform to the requirements specified in other Parts of the Standard.

An implementation may conform to the DICOM Web Services by supporting the role of origin server or user agent, or both, for any of the Services defined in this Part of the Standard. The structure of Conformance Statements is specified in PS3.2.

50 An implementation shall describe in its Conformance Statement the Real-World Activity associated with its use of DICOM Web Services, including any proxy functionality between a Web Service and the equivalent DIMSE Service.

An implementation shall describe in its Conformance Statement the security mechanisms utilized by the implementation. See Section 8.11.

Modify PS3.18 Section 7.2 as indicated

55 *(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):*

7.2 Resources, Representations, and Target URIs

60 In RESTful Web Services, a resource is an abstract object with a type, associated data, relationships to other resources, and a set of methods that operate on it. Resources are grouped into collections. Collections are themselves resources as well. Each collection is unordered and contains only one type of resource. Collections can exist globally, at the top level of an API, but can also be contained inside a resource. In the latter case, we refer to these collections as sub-collections. Sub-collections usually express some kind of "contained in" relationship.

Modify PS3.18 Section 8.6.1.2 as indicated

(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):

65 8.6.1.2 Multipart Payload

A message with a multipart payload contains zero or more representations. Each representation goes in a separate part.

A message with a multipart payload shall have a Content-Type Header Field with a multipart media-type.

70 The Media Type of the root representation (see [RFC2387]) may be specified by the Content-Type Header Field of the message. If no root parameter is specified, then the root representation is the first representation in the payload.

Each part in a multipart payload shall start with a boundary string, followed by a Content-Type Header Field with a single part Media Type (see Section 8.7.3), followed by other fields as specified in Table 8.6.1-1. See also Figure 8.6-1. Other Header Fields may be included.

Note

- 75
1. Understanding the nature of an encoded Bulkdata resource may depend on the corresponding Metadata reference to the ~~b~~**Bulk_d**~~Data~~**Data**_URI and is not necessarily implicit in the Content-Type Header Field.
 2. An HTTP Range [RFC7233] request may be used with Multipart payloads, but the range applies to the entire response, including the multipart markers. In order for the response to be valid across requests, the ordering of items and the choice of multipart separator must remain the same.

80 The Content-Location is used to identify the specific resource (e.g., down to the level of a specific Frame or instance or ~~b~~**Bulk_d**~~Data~~**Data**_URI) represented in this part. This allows the payload recipient to distinguish the parts, for example when each part contains a different frame of a requested Multi-frame Instance.

Note

- 85
1. The metadata in the response of a Search Transaction is not considered a representation of a resource, so a Content-Location is not required.
 2. In the case of a rendered resource, the Content-Location will identify the resource from which the rendering was generated.

Table 8.6.1-1. Multipart Header Fields

Name	Value	Usage	Description
Content-Type	media-type	M	
Content-Length	uint	C	Shall be present if the response payload does not have a transfer encoding
Content-Location	url	C	Shall be present if the response payload contains a representation of a resource. See [RFC7231] Section 3.1.4.2.
Location	url	C	See [RFC7231] Section 7.1.2.

90 See Section 8.7.1 and [RFC7231].

The following is an example template of a multipart request or response message that has a multipart payload:

```
request-line / response-line
Content-Type: multipart-media-type CRLF
Content-Location: "/" {/url} CRLF
```

95 *(header-field CRLF)

CRLF

multipart-payload

The Content-Type Header Field shall have a multipart media-type. For example:

100 Content-Type: multipart/related; type=DQUOTE root-media-type DQUOTE; boundary="---
boundary---"

Where

multipart-media-type is a Media Type defined by [RFC2387].

root-media-type is a single part Media Type that specifies the Media Type of the root, typically the first part, in the payload. If the value of the type parameter and the root body part's content-type differ then the user agent's behavior is undefined.

boundary specifies a string that acts as a boundary between message parts.

105 If a multipart payload contains representations of Metadata (see Section 8.7.3.3.1), and Bulk ~~d~~Data (see Section 8.7.3.3.2), then all Metadata message parts that reference a Bulk ~~d~~Data part shall precede the referenced Bulk ~~d~~Data part. The Content-Location of the Bulk ~~d~~Data part shall contain the corresponding ~~b~~Bulk_~~d~~Data_URI used in the referencing Metadata.

Figure 8.6-1 shows the correspondence between the IOD representation and a multipart payload.

[...]

110 **Modify PS3.18 Section 8.6.2 as indicated**

(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):

8.6.2 DICOM Representations

All DICOM ~~e~~Objects are defined by Information Object Definitions (IODs). See PS3.3. Representations of DICOM Resources are encodings of DICOM Information Objects into octet streams.

115 Each IOD has an associated set of Attributes, which define semantic concepts. Each Attribute has:

- a Tag, which identifies the Attribute using an integer
- a Keyword, which identifies the Attribute using a token
- a Type, which indicates whether the Attribute is required or optional
- a Value Representation, which defines the data type of the Attribute's value(s)
- a Value Multiplicity, which specifies the number of values that the Attribute may have

A Data Element is a concrete representation of an Attribute See PS3.5. Each Data Element has:

- an identifier, which would typically be its Tag, but could be its Keyword
- a Value Representation, which defines its data type
- a Value Length
- a Value Field, which is a sequence of bytes containing zero or more values

Each Instance contains Data Elements representing the Attributes from the Patient, Study, Series, and Instance levels of the IOD. For example, if a Series resource contains 12 Instances, then a transaction that retrieves that Series will contain a representation of the Series and its 12 Instances, in a specific Media Type, and each Instance will have Patient, Study, Series, and Instance level Attributes.

This Part of the Standard defines three distinct representations of DICOM Resources that can be encoded into DICOM Media Types: Instances, Metadata, and Bulk ~~d~~ Data.

DICOM Media Types and their corresponding representations are defined in Section 8.7.3. Other Media Types used in this Part of the Standard are defined in Section 8.7.4.

Modify PS3.18 Section 8.7.3 as indicated

(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):

8.7.3 DICOM Media Type Sets

This section defines the Media Types used to represent DICOM Instances, Metadata and Bulk ~~d~~ Data. It describes:

- The Media Type and Transfer Syntax parameters for DICOM PS3.10 Instances
- The Media Types that can be used for Metadata
- The Media Types and Transfer Syntaxes parameters for Bulk ~~d~~ Data
- The syntax of DICOM Media Types including their Transfer Syntax and character set parameters
- The Query Parameter for Transfer Syntax
- The meaning of Acceptable Transfer Syntaxes and Selected Transfer Syntax

The Media Types defined in this section are distinct from those into which DICOM Instances may be rendered (which are defined in Section 8.7.4); some of the same Media Types are used for both rendered content and Bulk ~~d~~ Data.

Depending on the service, the Media Types may be single part or multipart, and may have required or optional Transfer Syntax and/or character set parameters.

The Implicit VR Little Endian (1.2.840.10008.1.2), and Explicit VR Big Endian (1.2.840.10008.1.2.2 - Retired) Transfer Syntaxes shall not be used with Web Services.

If a Transfer Syntax parameter for a DICOM Media Type is not specified in a request or response, the Transfer Syntax in the response shall be the Transfer Syntax specified as the default for the Resource Category and Media Type combination in Table 8.7.3-2, Table 8.7.3-4 or Table 8.7.3-5, 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.

Table 8.7.3-1 specifies the definition of Media Type requirement terms used in the tables in this section.

[...]

Modify PS3.18 Section 8.7.3.3 as indicated

(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):

8.7.3.3 Bulkdata Media Types

Bulkd Data representations are only supported by RESTful services. There are two categories of Bulkd Data: 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 Section 8.7.8, 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 Bulkd Data.

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

- one compressed Bulkd Data representation, or
- one uncompressed Bulkd Data 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 Bulkd Data representations: one for each frame, or
- one Multi-Frame Image uncompressed Bulkd Data representation.

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

- one Video compressed Bulkd Data representation, or
- one Video uncompressed Bulkd Data representation.

8.7.3.3.1 Uncompressed Bulkd Data Media Types

Table 8.7.3-4 specifies the default Media Type and Transfer Syntax UIDs, by Resource Category (see Table 8.7.2-1) that can be used with uncompressed Bulkd Data for the RESTful services. Uncompressed Bulkd Data 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 Bulkd Data.
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.

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

Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	RESTful
Single Frame Image	application/octet-stream	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
Multi-Frame Image	application/octet-stream	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
Video	application/octet-stream	1.2.840.10008.1.2.1	Explicit VR Little Endian	D
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

205 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 Bulk Data.

8.7.3.3.2 Compressed Bulk Data Media Types

Compressed Bulk Data contains only the compressed octet stream without the fragment delimiters.

210 Table 8.7.3-5 specifies the default and optional Media Types and Transfer Syntax UID combinations for each Resource Category (see Table 8.7.2-1) of compressed Bulk Data for the RESTful services.

Note

- 215
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 Bulk Data, such as images or video, encoded in a specific Transfer Syntax.

220 For details on how Compressed Bulk Data is packaged into single part or multipart payloads, see Section 8.6.1.

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

Resource Category	Media Type	Transfer Syntax UID	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
	image/dicom-rle	1.2.840.10008.1.2.5	RLE Lossless	D

Resource Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	Optionality
	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
	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

Resource Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	Optionality
	image/jphc	1.2.840.10008.1.2.4.93	JPEG 2000 Part 2 Multi-component Image Compression	O
		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

Resource Category	Media Type	Transfer Syntax UID	Transfer Syntax Name	Optionality
		1.2.840.10008.1.2.4.108	HEVC/H.265 Main 10 Profile / Level 5.1	O
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.

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

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

235 Compressed multi-frame image pixel data is encoded as individual frames. E.g., each frame of a JPEG 2000 multi-frame image will be encoded separately as image/jp2 representations, rather than as a single video/mj2 ([RFC3745]) or application/octet-stream representation. See Section 8.6.1.2 for details on how multiple representations can be packaged into a multipart payload.

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 ([RFC4337]) representation.

Note

240 1. The resource on the origin server may have been encoded in the Deflated Explicit VR Little Endian (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 Transfer Syntax. Alternatively, if the user agent allowed a Content-Encoding Header Field of 'deflate', then the deflated bytes may be transferred unaltered, but the Transfer Syntax parameter in the response should be the Explicit VR Little Endian Transfer Syntax.

245 2. Many of the Media Types used for compressed Pixel Data transferred as Bulk d Data values are also used for consumer format Media Types. A web browser may not be able to display the encoded data directly, even though some of the same Media Types are also used for encoding rendered Pixel Data. See Section 8.7.4.

250 For example, the Media Type for Bulk d Data values of lossless 16-bit JPEG [ISO/IEC 10918-1] encoded Pixel Data is "image/jpeg", the same Media Type as might be used for 8-bit JPEG [ISO/IEC 10918-1] encoded Pixel Data, whether extracted as Bulk d Data, or rendered. The Transfer Syntax parameter of the Content-Type Header Field is useful to signal the difference.

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

Modify PS3.18 Section 8.7.4 as indicated

(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):

8.7.4 Rendered Media Types

260 DICOM Instances may be converted by a rendering process into non-DICOM Media Types. This can be useful to display or process them using non-DICOM software, such as browsers.

For example, an Instance containing:

1. an image could be rendered into the image/jpeg, image/jph, image/jxl, image/png, or image/gif Rendered Media Types.

- 265 2. a multi-frame image in a lossless Transfer Syntax could be rendered into a video/mpeg, video/mp4, video/H265, or image/jxl Rendered Media Type.
3. a Structured Report could be rendered into a text/html, text/plain, or application/pdf Rendered Media Type.

Note

270 Rendered Media Types are usually consumer format Media Types. Some of the same non-DICOM Media Types are also used as Bulkdata Media Types, that is, for encoding Bulk Data extracted from Encapsulated Pixel Data (used with compressed Transfer Syntaxes), without applying a rendering process. See Section 8.7.3.3.

The rendering of Presentation States is specified in Section 8.3.5.1.

275 Origin servers shall support rendering Instances of different Resource Categories into Rendered Media Types as specified in Table 8.7.4-1.

Modify PS3.18 Section 8.7.8.1 as indicated

(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):

8.7.8.1 Selected Media Type

280 The Selected Media Type is the Media Type selected by the origin server for the representation in the response payload. The Media Types in the Accept Query Parameter and the media ranges in the Accept Header Field shall each be separately prioritized according to the rules defined in [RFC7231] Section 5.3.1.

For multipart payloads, the Selected Media Type is determined independently for each message part in the response.

285 The Selected Media Type of each message part depends on the Resource Category of the Instance and the Acceptable Media Types for that Resource Category.

The Selected Media Type is chosen as follows:

1. Identify the target's Resource Category
2. Select the representation with the highest priority supported Media Type for that category in the Accept Query Parameter.
- 290 3. If no Media Type in the Accept Query Parameter is supported, select the highest priority supported Media Type for that category in the Accept Header Field, if any.
4. Otherwise, select the default Media Type for the category, if the Accept Header Field contains a wildcard media range matching the category, if any.
5. Otherwise, return a 406 (Not Acceptable).

295 Note

1. If the Selected Media Type is the Explicit VR Little Endian and the pixel data is compressed and when uncompressed is of such length that it cannot be contained in a ~~Value Field~~^{Value Field}, then the origin server will respond with a 406 (Not Acceptable), and the user agent may try again with a different set of Acceptable Media Types.
- 300 2. If transcoding to the Explicit VR Little Endian Transfer Syntax, a VR of UN may be needed for the encoding of Data Elements with explicit VR whose value length exceeds 65534 ($2^{16}-2$) (FFFEH, the largest even length unsigned 16-bit number) but which are defined to have a 16-bit explicit VR length field. See Section 6.2.2 "Unknown (UN) Value Representation" in PS3.5.

305 For a set of Media Types in the Accept Query Parameter (step 2 above), or for a set of media ranges in the Accept Header Field (step 3 above), the highest priority supported Media Type is determined as follows:

1. Assign a qvalue of 1 to any member of the set that does not have a one.
2. Assign each representation supported by the origin server the qvalue of the most specific Media Type that it matches.
3. Select the representation with the highest qvalue. If there is a tie, the origin server shall determine which is returned.

310 For example, consider an origin server which receives a request with the following Accept Header Field:

```
Accept: text/*; q=0.5, text/html; q=0.4, text/html; level=1, text/html; level=2;
q=0.7,
        image/png, */*; q=0.4
```

315 Suppose that for the resource indicated in the request, the origin server supports representations for the following Media Types:

```
text/html (regular, level 1 and level 2)
text/rtf
text/plain
text/x-latex
```

320 These Media Types are assigned the following qvalues, based on the media ranges above:

[...]

Modify PS3.18 Section 9.5.2.1 as indicated

325 *(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):*

9.5.2.1 Frame Number

If this Query Parameter is supported and is present in the request, the origin server shall use the specified frame as the Target Resource.

However, if any of the following are true:

- 330
- the Target Resource is not a multi-frame image or video,
 - the number of parameter values is not equal to one, or
 - the parameter value is not a positive integer less than or equal to the number of frames in the Instance

the origin server shall return a 400 (Bad Request) response and may include a payload containing an appropriate error message.

335

Modify PS3.18 Chapter 10 as indicated

(changes to existing text are bold and underlined for additions and bold and struckthrough for removals):

10 Studies Service and Resources

10.1 Overview

340 The Studies Resource enables a user agent to store, retrieve, update, and search an origin server for DICOM Studies, Series, and Instances, along with their /metadata, /rendered, and /thumbnail variants; as well as Frames and Bulk Data.

The Retrieve transaction of this Service is also known as WADO-RS. The Store transaction of this Service is also known as STOW-RS. The Search transaction of this Service is also known as QIDO-RS. See Section 10.3.

345 10.1.1 Resource Descriptions

The Studies Service manages a collection of DICOM Study resources. Each Study is organized in a hierarchy of sub-resources that correspond to the DICOM Information Model. See Section 7 “DICOM Model of the Real World” in PS3.3.

There are three top level resources:

/studies	references all Studies managed by the service.
/series	references all Series managed by the service.
/instances	references all Instances managed by the service.

350 The following URI Template variables are used in resource definitions in this Section.

{study}	the Study Instance UID of a Study managed by the Studies Service.
{series}	the Series Instance UID of a Series contained within a Study resource.
{instance}	the SOP Instance UID of an Instance contained within a Series resource.
{frames}	a comma-separated list of frame numbers, in ascending order, contained within an Instance.
{bulkdataURI}	an opaque URI that references a Bulk <u>d</u> Data Value.

The Studies Service defines the following resources:

Table 10.1-1. Resources and Descriptions

Resource	Description
Studies Service	The Base URI of the Studies Service.
All Studies	The All Studies resource references the entire collection of Studies contained in the Studies Service.
Study	The Study resource references a single Study.
Study Metadata	The Study Metadata resource references the Metadata of a Study.
Study Bulk <u>d</u> Data	The Study Bulk <u>d</u> Data resource references the Bulk <u>d</u> Data of a Study.
Study Pixel Data	The Study Pixel Data resource references the Pixel Data of a Study.
Rendered Study	The Rendered Study resource references an alternate Media Type rendering of a Study.
Rendered MPR Volume Study	The Rendered MPR Volume Study resource references a multiplanar reformat rendering of a Study.
Rendered 3D Volume Study	The Rendered 3D Volume Study resource references a volume rendering of a Study.
Study Thumbnail	The Study Thumbnail resource references a thumbnail image of a Study.
Study's Series	The Study's Series resource references the collection of all Series contained in a Study.
Study's Instances	The Study's Instances resource references the collection of all Instances in a single Study.
All Series	The All Series resource references the collection of all Series in all Studies contained in the Studies Service.
Series	The Series resource references a single Series.
Series Metadata	The Series Metadata resource contains the Metadata of a Series in a Study.

Resource	Description
Series Bulk <u>d</u> Data	The Series Bulk <u>d</u> Data resource references the Bulk <u>d</u> Data of a Series.
Series Pixel Data	The Series Pixel Data resource references the Pixel Data of a Series.
Rendered Series	The Rendered Series resource references an alternate Media Type rendering of a Series.
Rendered MPR Volume Series	The Rendered MPR Volume Series resource references a multiplanar reformat rendering of a Series.
Rendered 3D Volume Series	The Rendered 3D Volume Series resource references a volume rendering of a Series.
Series Thumbnail	The Series Thumbnail resource references a thumbnail image of a Series.
Series' Instances	The Series' Instances resource references the collection of all Instances in a single Series.
All Instances	The All Instances resource references the collection of all Instances in all Series in all Studies contained in the Studies Service.
Instance	The Instance resource references a single Instance.
Instance Metadata	The Instance Metadata resource contains the Metadata of an Instance.
Instance Bulk <u>d</u> Data	The Instance Bulk <u>d</u> Data resource references the Bulk <u>d</u> Data of an Instance.
Instance Pixel Data	The Instance Pixel Data resource references the Pixel Data of an Instance.
Rendered Instance	The Rendered Instance resource references an alternate Media Type rendering of an Instance.
Rendered MPR Volume Instance	The Rendered MPR Volume Instance resource references a multiplanar reformat rendering of an Instance.
Rendered 3D Volume Instance	The Rendered 3D Volume Instance resource references a volume rendering of an Instance.
Instance Thumbnail	The Instance Thumbnail resource references a thumbnail image of an Instance.
Frames	The Frames resource references an ordered collection of frames in a single multi-frame Instance.
Rendered Frames	The Rendered Frames resource references an alternate Media Type rendering of an ordered collection of frames of a single multi-frame Instance.
Rendered MPR Volume Frames	The Rendered MPR Volume Frames resource references a multiplanar reformat rendering of a collection of frames.
Rendered 3D Volume Frames	The Rendered 3D Volume Frames resource references a volume rendering of a collection of frames.
Frame Thumbnail	The Frame Thumbnail resource references a thumbnail image for frames within an Instance.
Bulk <u>d</u> Data	The Bulk <u>d</u> Data resource contains a Bulk <u>d</u> Data Value.

355 Note

There is no Frame BulkdData or Frame Pixel Data resource because they would be equivalent to the Frames resource.

[...]

360 F.2.2 DICOM JSON Model Object Structure

....

Each attribute object contains the following named child objects:

- ...

- At most one of:

365 • ...

- BulkDataURI: A string encoding the WADO-RS URL of a ~~bulk data item~~ **Bulk Data Value** describing the Value Field of an enclosing Attribute with a VR of DS, FL, FD, IS, LT, OB, OD, OF, OL, OV, OW, SL, SS, ST, SV, UC, UL, UN, US, UT or UV (described in Section F.2.6)

370 • InlineBinary: A base64 string encoding the Value Field of an enclosing Attribute with a VR of OB, OD, OF, OL, OV, OW, or UN (described in Section F.2.7)

...

F.2.6 BulkDataURI

If an attribute contains a "BulkDataURI", this contains the URI of a ~~bulk data element~~ **BulkData Element Value** as defined in Table A.1.5-2 in PS3.19.

375 ***For reference PS3.19 (unchanged)***

Table A.1.5-2. DICOM Data Set Macro

Name	Optionality	Cardinality	Description
DicomAttribute	O	0-n	An InfoSet element corresponding to each DICOM Attribute.
...			
>BulkData	C	1	<p>A reference to a blob of data that the recipient may retrieve through use of the GetData() method, a PS3.18 Studies Service Retrieve (WADO-RS) transaction or a PS3.18 Studies Service Store (STOW-RS) transaction.</p> <p>Required if the DICOM Data Element represented is not zero length and an XML InfoSet Value, Item, InlineBinary or PersonName element is not present.</p> <p>The provider of the data may use a BulkData reference at its discretion to avoid encoding a large DICOM Value Field as text by value in the InfoSet. For example, pixel data or look up tables.</p> <p>There is a single BulkData InfoSet element representing the entire Value Field, and not one per Value in the case where the Value Multiplicity is greater than one.</p> <p>Note</p> <p>E.g., a LUT with 4096 16 bit entries that may be encoded in DICOM with a Value Representation of OW, with a VL of 8192 and a VM of 1, or a US VR</p>

Name	Optionality	Cardinality	Description
			<p>with a VL of 8192 and a VM of 4096 would both be represented as a single BulkData element.</p> <p>All rules (e.g., byte ordering and swapping) in PS3.5 apply.</p> <p>Note</p> <p>Implementers should pay particular attention to the PS3.5 rules regarding the value representations of OD, OF, OL, OV and OW.</p> <p>If the BulkData has a string or text Value Representation, the value(s) of the DICOM Specific Character Set Data Element, if present, might be necessary to determine its encoding.</p>
>>uuid	C	A	<p>An identifier of this bulk data reference formatted as a UUID using the hexadecimal representation defined in [ITU-T X.667].</p> <p>Required if BulkData URI is not present. Shall not be present otherwise.</p>
>>uri	C	A	<p>The HTTP(S) URI for this bulk data reference.</p> <p>Required if the NativeDicomModel was:</p> <ul style="list-style-type: none"> • returned in response to a PS3.18 Studies Service Retrieve (WADO-RS) Retrieve Metadata request <p>Shall not be present otherwise.</p>
>InlineBinary	C	1	<p>The Value Field of the enclosing Attribute encoded as base64.</p> <p>Required if the DICOM Data Element represented is:</p> <ul style="list-style-type: none"> • not zero length • the VR if the enclosing Attribute is OB, OD, OF, OL, OV, OW, or UN • an XML InfoSet Value or BulkData XML element is not present <p>Shall not be present otherwise.</p> <p>There is a single InlineBinary InfoSet element representing the entire Value Field, and not one per Value in the case where the Value Multiplicity is greater than one.</p> <p>Note</p> <p>E.g., a LUT with 4096 16 bit entries that may be encoded in DICOM with a Value Representation of OW with a VL of 8192 and a VM of 1 would be represented as a single InlineBinary element.</p> <p>All rules (e.g., byte ordering and swapping) in PS3.5 apply.</p>

Name	Optionality	Cardinality	Description
			<p>Note</p> <p>Implementers should pay particular attention to the PS3.5 rules regarding the value representations of OD, OF, OL, OV and OW.</p>