

1	Status	Final Text
2	Date of Last Update	2018/11/10
3	Person Assigned	David Clunie
4		mailto:dclunie@dclunie.com
5	Submitter Name	Shabir Adeelmailto:sadeel@google.com
6		Gunter Zeilinger mailto:gunterze@gmail.com
7	Submission Date	2018/03/08

8	Correction Number CP-1791	
9	Log Summary: Clarify Transfer Syntax for STOW-RS of PS3.10 files	
10	Name of Standard	
11	PS3.18 2018d	
12	Rationale for Correction:	
13	The Transfer Syntax for STOW-RS Requests to store PS3.10 files is derived from the meta information in the file and not required	
14	as a media Type parameter, as it is for separately encoded bulk data. Clarify this with a note.	
15	Correction Wording:	

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

6.6.1.1 Request

The specific Service resource to be used for the Store Instances action shall be as follows:

- Resource
 - {SERVICE}/studies/{StudyInstanceUID}, where
 - {SERVICE} is the base URL for the service. This may be a combination of scheme (either HTTP or HTTPS), host, port, and application.
 - {StudyInstanceUID} (optional) is the study instance UID for a single study. If not specified, instances can be from multiple studies. If specified, all instances shall be from that study; instances not matching the StudyInstanceUID shall be rejected.

Note

It is not necessary that the study referenced by the StudyInstanceUID in the resource (and in the provided instances) exists on the server, however it is necessary that it be a valid UID. The client may have obtained an appropriate UID from elsewhere or generated it as described in ??? and ???.

- Method
 - POST
- Headers
 - Content-Type - The representation scheme being posted to the RESTful service. The types allowed for this request header are as follows:
 - multipart/related; type="application/dicom"; boundary={messageBoundary}
Specifies that the post is ??? binary instances. All STOW-RS providers shall accept this Content-Type.
 - multipart/related; type="application/dicom+xml"; boundary={messageBoundary}
Specifies that the post is ??? XML metadata and bulk data. All STOW-RS providers shall accept this Content-Type.
 - multipart/related; type="application/dicom+json"; boundary={messageBoundary}
Specifies that the post is DICOM JSON metadata and bulk data. All STOW-RS providers shall accept this Content-Type.

6.6.1.1.1 DICOM Request Message Body

The DICOM Request Message has a multipart body.

- Content-Type:
 - multipart/related; type="application/dicom"; boundary={MessageBoundary}
- The multipart request body contains every instance to be stored. Each instance is in a separate part of the multipart body.
- Each part in the multipart body represents a DICOM SOP Instance with the following HTTP headers:
 - Content-Type: application/dicom

Note

The Transfer Syntax of each instance stored is encoded in Transfer Syntax UID (0002.0010) of the PS3.10 File Meta Information, and is not repeated in the Content-Type HTTP header as a parameter. If a transfer-syntax parameter is present and inconsistent with the PS3.10 File Meta Information, it will either be ignored or an error will be returned.

6.6.1.1.2 XML Metadata and Bulk Data Request Message Body

The XML Metadata and Bulk Data Request Message has a multipart body.

- Content-Type:
 - multipart/related; type="application/dicom+xml"; boundary={MessageBoundary}
- The multipart request body contains all the metadata and bulk data to be stored. If the number of bulk data parts does not correspond to the number of unique BulkDataURIs in the metadata then the entire message is invalid and will generate an error status line.
- Each body part is either DICOM XML metadata or a bulk data item from a SOP Instance sent as part of the Store operation. The first part of the multipart message shall be XML metadata.
- Each bulk data item shall be preceded by all metadata items that contain a reference to it.

Note

This requires that all bulk data items for an instance shall be preceded by the XML metadata for that instance and if a bulk data item is included in multiple instances it shall be preceded by the XML metadata for each instance in which it is included.

- The first part in the multipart request will contain the following HTTP headers:
 - Content-Type: application/dicom+xml; transfer-syntax={TransferSyntaxUID}
- Subsequent items will contain the following HTTP headers (order is not guaranteed):
 - additional metadata with the following headers:
 - Content-Type: application/dicom+xml; transfer-syntax={TransferSyntaxUID}

Where {TransferSyntaxUID} is the UID of the DICOM Transfer Syntax used to encode the inline binary data in the XML metadata.
 - an encapsulated document with the following headers:
 - Content-Type: {media-type}
 - Content-Location: {BulkDataURI}
 - an uncompressed bulk data element encoded in Little Endian binary format with the following headers:
 - Content-Type: application/octet-stream
 - Content-Location: {BulkDataURI}
 - a compressed pixel data object from a SOP Instance in the Study with the following headers:
 - Content-Type: {media-type} [dcm-parameters]
 - Content-Location: {BulkDataURI}
- Metadata and its associated bulk data shall always be sent in the same POST request.

Note

It is not intended that metadata and bulk data be stored separately in multiple POST requests since the service always requires the metadata for context.

6.6.1.1.3 JSON Metadata and Bulk Data Request Message Body

The JSON Metadata and Bulk Data Request Message has a multipart body.

- Content-Type:

- 1 • multipart/related; type="application/dicom+json"; boundary={MessageBoundary}
- 2 • The multipart request body contains all the metadata and bulk data to be stored. If the number of bulk data parts does not correspond
- 3 to the number of unique BulkDataURIs in the metadata then the entire message is invalid and will generate an error status line.
- 4 • The first part in the multipart request will contain a JSON array of DICOM JSON Model Objects (defined in ???). Each array element
- 5 is the metadata from a SOP Instance sent as part of the Store operation. This message part will have the following headers:
 - 6 • Content-Type: application/dicom+json; transfer-syntax={TransferSyntaxUID}
 - 7 Where {TransferSyntaxUID} is the UID of the DICOM Transfer Syntax used to encode the inline binary data in the JSON metadata.
- 8 • Subsequent items will be one of the following:
 - 9 • an encapsulated document with the following headers:
 - 10 • Content-Type: {media-type}
 - 11 • Content-Location: {BulkDataURI}
 - 12 • an uncompressed bulk data element encoded in Little Endian binary format with the following headers:
 - 13 • Content-Type: application/octet-stream
 - 14 • Content-Location: {BulkDataURI}
 - 15 • a compressed pixel data object from a SOP Instance in the Study with the following headers:
 - 16 • Content-Type: {media-type} [dcm-parameters]
 - 17 • Content-Location: {BulkDataURI}
- 18 • JSON Metadata and its associated bulk data shall always be sent in the same POST request.

Note

19 It is not intended that metadata and bulk data be stored separately in multiple POST requests since the service always requires

20 the metadata for context.

21