

DICOM Correction Proposal

STATUS	Final Text
Date of Last Update	2014/06/25
Person Assigned	James Philbin
Submitter Name	Jonathan Whitby (jwhitby@vitalimages.com)
Submission Date	2013/10/17

Correction Number	CP-1351
Log Summary: Add JSON support to WADO-RS RetrieveMetadata and STOW-RS services	
Name of Standard PS 3 2013	
Rationale for Correction: DICOM JSON is defined in Sup 166 as an alternative representation of DICOM Data Sets intended primarily for use with mobile and web clients. An expected QIDO-RS use case is to search for Studies, Series or Instances and then use WADO-RS to retrieve them. It would be useful to not only allow QIDO-RS search results in JSON but also WADO-RS metadata. Additionally, a mobile client may want to upload data using JSON metadata. NOTE: This CP depends on Sup 166.	
Correction Wording:	
Update PS 3.18 Section 6.5 RS REQUEST/RESPONSE as indicated.	

6.5 RS REQUEST/RESPONSE

The DICOM RESTful Service defines several action types. An implementation shall support all the following six action types:

1. RetrieveStudy
This action retrieves the set of DICOM instances associated with a given study unique identifier (UID). The response can be DICOM or bulk data depending on the "Accept" type, and is encapsulated in a multipart MIME response.
2. RetrieveSeries
This action retrieves the set of DICOM instances associated with a given study and Series Instance UID. The response can be DICOM or bulk data depending on the "Accept" type, and is encapsulated in a multipart MIME response.
3. RetrieveInstance
This action retrieves the DICOM instance associated with the given study, series, and SOP Instance UID. The response can be DICOM or bulk data depending on the "Accept" type, and is encapsulated in a multipart MIME response.
4. RetrieveFrames

This action retrieves the DICOM frames for a given study, series, SOP Instance UID, and frame numbers. The response is pixel data, and encapsulated in a multipart MIME response.

5. RetrieveBulkdata

This action retrieves the bulk data for a given bulk data URL. The response is a single bulk data item.

6. RetrieveMetadata

This action retrieves the DICOM instances presented as the full study metadata with the bulk data removed. ~~The response is XML encoded metadata for the DICOM attributes as defined in PS 3.19.~~

~~All responses will be http multipart messages.~~

DICOM objects returned shall be PS_3.10 binary objects encoded in a requested Transfer Syntax (Explicit VR Little Endian by default) with one message part per DICOM Instance.

...

Other types of responses will be encoded in the following manner: (see Figure 6.5-1).

- All XML responses shall be encoded as described in the Native DICOM Model defined in PS3.19 with one message part per XML object.
- All JSON responses shall be encoded as a DICOM JSON Model Object as defined in Annex F.
- Uncompressed bulk and pixel data shall be encoded in a Little Endian format using the application/octet-stream media type with one message part per bulk data item.
- ...

Update PS 3.18 Section 6.5.1.2 Response as indicated.

6.5.1.2 Response

The Server shall provide the document(s) indicated in the request. In order to parse the bulk data items it is necessary to also retrieve the ~~XML~~ metadata for the Study.

Update PS 3.18 Section 6.5.2.2 Response as indicated.

6.5.2.2 Response

The Server shall provide the document(s) indicated in the request. In order to parse the bulk data items it is necessary to also retrieve the ~~XML~~ metadata for the Study.

Update PS 3.18 Section 6.5.3.2 Response as indicated.

6.5.3.2 Response

The Server shall provide either a single DICOM PS3.10 object for the SOP Instance or one or more bulk data items. In order to parse the bulk data items it is necessary to also retrieve the ~~XML~~ metadata for the

Study.

Update PS 3.18 Section 6.5.4.2 Response as indicated.

6.5.4.2 Response

The Server shall provide the document(s) indicated in the request. In order to parse the bulk data items it is necessary to also retrieve the ~~XML~~ metadata for the Study.

Update PS 3.18 Section 6.5.5.1 Response as indicated.

6.5.5.1 Request

The specific Services resource to be used for the RetrieveBulkdata action shall be as follows:

- Resource
 - {BulkDataURL}, where
 - {BulkDataURL} is the URL of a bulk data element. This may be the URL attribute of a BulkData element ~~from a DICOM PS3.19 XML file~~ received in response to a WADO-RS RetrieveMetadata request.

Update PS 3.18 Section 6.5.5.2 Response as indicated.

6.5.5.2 Response

The Server shall provide the document(s) indicated in the request. In order to parse the bulk data items it is necessary to also retrieve the ~~XML~~ metadata for the Study.

Update PS 3.18 Section 6.5.6 RS – RetrieveMetadata as indicated.

6.5.6 RS – RetrieveMetadata

This action retrieves the DICOM instances presented as the full study metadata with the bulk data removed. The response is ~~XML encoded~~ metadata for the DICOM attributes ~~as defined in PS 3.19~~.

...

6.5.6.1 Request

...

- Headers
 - Accept
 - multipart/related; type=application/dicom+~~xml~~
Specifies that the response should be WADO PS 3.19 XML. All WADO-RS providers must support this Media Type.
 - application/json
Specifies that the results should be DICOM JSON (see Annex F). A WADO-RS provider may optionally support this Media Type

6.5.6.2 Response

The Server shall provide the document(s) indicated in the request. The Server shall return the document(s) or an error code when the document(s) could not be returned.

The response ~~format~~ has a content type of either:

- multipart/related; type=application/dicom+xml, as described in the Native DICOM Model defined in PS3.19;

or

- application/json as described in Appendix F.

The response must include the URL attribute for each BulkData element.

6.5.6.2.1 XML Metadata Response

- Content-Type:
 - multipart/related; type=application/dicom+xml
- The entire multipart response contains all XML metadata for the specified Study.
- Each item in the response is the XML encoded metadata for an Instance with the following http 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.

6.5.6.2.2 JSON Metadata Response

- Content-Type:
 - application/json; transfer-syntax={TransferSyntaxUID}
Where {TransferSyntaxUID} is the UID of the DICOM Transfer Syntax used to encode the inline binary data in the metadata.

The response is a JSON array that contains all metadata for the specified Study.

Each element in the array is the DICOM JSON encoded metadata for an Instance (see Annex F)

Note

The metadata is consistent with the characteristics of the bulk data on the server. If bulk data is requested using specified Transfer Syntaxes or media types, it is possible that the bulk data retrieved may be inconsistent with the metadata. For example, for a Study whose DICOM Tag (0028,2110) "LossyImageCompression" is set to "00", indicating no lossy compression, calling RetrieveStudy and requesting a lossy compression media type will provide pixel data that is inconsistent with the metadata. It is the responsibility of the client to deal with these inconsistencies appropriately.

Update PS 3.18 Section 6.6 RS REQUEST/RESPONSE as indicated

6.6 STOW-RS REQUEST/RESPONSE

The STOW-RS Service defines one action type. An implementation shall support the following action type:

1. Store Instances

This action creates new resources for the given SOP Instances on the Server or appends to existing resources on the Server.

All request messages are HTTP/1.1 multipart messages. The organization of SOP Instances into message parts depends on whether the SOP Instances are structured as PS 3.10 binary instances, or metadata and bulk data.

PS 3.10 binary instances shall be encoded with one message part per DICOM Instance.

Metadata and bulk data requests will be encoded in the following manner:
(see Figure 6.5-1 WADO-RS Mapping **between** IOD and HTTP message body parts)

- All XML request messages shall be encoded as described in the Native DICOM Model defined in PS 3.19 with one message part per XML object.
- **All JSON requests shall be encoded as an array of DICOM JSON Model Objects defined in Annex F.**
- Uncompressed bulk and pixel data shall be encoded in a Little Endian format using the application/octet-stream media type with one message part per bulk data item.
- Compressed pixel data shall be encoded in one of two ways:
 - Single-frame pixel data encoded using a single-frame media type (one message part)
 - Multi-frame or video pixel data encoded using a multi-frame media type (multiple frames in one message part)

Update PS 3.18 Section 6.6.1.1 Request as indicated
--

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.
- 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 PS 3.10 binary instances. **All STOW-RS providers must accept this Content-Type.**
- multipart/related; type=application/dicom+xml; boundary={messageBoundary}
Specifies that the post is PS 3.19 XML metadata and bulk data. **All STOW-RS providers must accept this Content-Type.**
- **multipart/related; type=application/json; boundary={messageBoundary}**
Specifies that the post is DICOM JSON metadata and bulk data. A STOW-RS provider may optionally accept this Content-Type.

...

Update PS 3.18 Section 6.6.1.1.2 Metadata and Bulk Data Request Message Body as indicated
--

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.

Append to PS 3.18 Section 6.6.1.1 Request as indicated

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:
 - multipart/related; type=application/json; 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.
- The first part in the multipart request will contain a JSON array of DICOM JSON Model Objects (defined in Annex F). Each array element is the metadata from a SOP Instance sent as part of the Store operation. This message part will have the following headers:
 - Content-Type: application/json; transfer-syntax={TransferSyntaxUID}
- Subsequent items will be one of the following:
 - 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: {MediaType}
 - Content-Location: {BulkDataURI}
- JSON 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.

Update PS 3.18 (2013) Section 6.6.1.3 Response

The RESTful Service shall return an HTTP status line, including a status code and associated textual phrase for the entire set of stored SOP Instances, followed by an ~~XML~~ message body containing ~~a PS3.19 XML representation of~~ the Store Instances Response Module as defined in Table 6.6.1-2. The message body shall be encoded as either:

- an XML object as described in the Native DICOM Model defined in PS 3.19,

or

- a DICOM JSON Model Object defined as defined in Annex F.

Update PS 3.18 6.6.1.3.1 Response Status Line as indicated

6.6.1.3.1 Response Status Line

If the status for all instances included in the POST request is Success, the RESTful Service shall return an “HTTP 200 – Success” response code.

If the status for all instances included in the POST request is Failure, the RESTful Service shall return an appropriate failure status line with a response code from Table 6.6.1-1. If there are instance specific errors, the response code shall be a 409 and the response payload shall contain the will be followed by the PS 3.18 XML Store Instances Response Module as the message body, which contains additional information regarding instance errors.

In all other conditions, the RESTful Service shall return an “HTTP 202 – Accepted” response code. The response payload may contain a Store Instances Response Module, which specifies A additional information regarding ~~the~~ instance warnings or failures. will be found in the PS 3.18 XML Store Instances Response Module in the response message body.

Table 6.6.1-1

HTTP/1.1 STANDARD RESPONSE CODE

Service Status	HTTP/1.1 Status Codes	STOW-RS Description
Failure	400 – Bad Request	This indicates that the STOW-RS Service was unable to store any instances due to bad syntax.
	401 – Unauthorized	This indicates that the STOW-RS Service refused to create or append any instances because the client is not authorized.
	403 – Forbidden	This indicates that the STOW-RS Service understood the request, but is refusing to fulfill it (e.g. an authorized user with insufficient privileges).
	409 – Conflict	This indicates that the STOW-RS Service request was formed correctly but the service was unable to store any instances due to a conflict in the request (e.g. unsupported SOP Class or StudyInstanceUID mismatch).

		<p>This may also be used to indicate that a STOW-RS Service was unable to store any instances for a mixture of reasons.</p> <p>Additional information regarding the instance errors can be found in the XML response message body.</p>
	<u>415 – Unsupported Media Type</u>	<u>This indicates that the STOW-RS Service does not support the Content-Type specified in the storage request (e.g., the service does not support JSON metadata)</u>
	503 – Busy	This indicates that the STOW-RS Service was unable to store any instances because it was out of resources.
Warning	202 – Accepted	<p>This indicates that the STOW-RS Service stored some of the instances but warnings or failures exist for others.</p> <p>Additional information regarding this error can be found in the XML response message body.</p>
Success	200 – OK	This indicates that the STOW-RS Service successfully stored all the instances.

Note: HTTP Status Codes for Failures and Warnings are returned in HTTP response headers. It is recommended that the text returned in the HTTP Response Warning contain a DICOM Status Code and descriptive reason as defined in Section 6.6.1.3.3. For example,

Warning: "A700: Out of memory"

Update PS 3.18 Section 6.6.1.3.2 Response Message Body

The ~~XML~~ message body shall provide appropriate status codes for individual SOP Instances indicating success, warning, or failure as defined below.

The ~~XML~~ message body may also include details about the processing of attributes by the service.

Table 6.6.1-2 defines the Attributes for referencing SOP Instances that are contained in a ~~PS3.18 XML~~ Store Instances Response Module in the response message body.

...

Update PS 3.18 Section 7.1.2 MIME type constraints as indicated.

7.1.2 MIME type constraints

The Server shall be able to send a response for each of the following MIME types:

- WADO-WS
 - application/dicom
 - image/jpeg

- WADO-RS
 - application/dicom
 - application/octet-stream
 - application/dicom+xml
 - application/json

Update PS 3.18 Section 7.2.2 MIME type constraints as indicated.

7.2.2 MIME type constraints

The Server shall be able to send a response for the following MIME types:

- WADO-WS
 - application/dicom
- WADO-RS
 - application/dicom
 - application/octet-stream
 - application/dicom+xml
 - application/json