

# **PS3.18**

## **DICOM PS3.18 2014c - Web Services**

## **PS3.18: DICOM PS3.18 2014c - Web Services**

Copyright © 2014 NEMA

# Table of Contents

Notice and Disclaimer .....	13
Foreword .....	15
1. Scope .....	17
2. Conformance .....	19
3. Normative References .....	21
4. Terms and Definitions .....	23
4.1. DICOM Persistent Object .....	23
4.2. Origin-Server .....	23
4.3. User-Agent .....	23
4.4. Web Client System .....	23
4.5. Web Enabled DICOM Server .....	23
4.6. Web Access to DICOM Persistent Objects .....	23
5. Symbols and Abbreviated Terms .....	25
6. Data Communication Requirements .....	27
6.1. Interaction .....	27
6.2. WADO-URI Request .....	27
6.2.1. Parameters of the HTTP Request .....	28
6.2.2. List of Media Types Supported in the Response .....	28
6.2.3. List of Character Sets Supported in the Response .....	28
6.3. WADO-URI Response .....	28
6.3.1. Body of Single DICOM MIME Sub-type Part Response .....	28
6.3.1.1. MIME Type .....	28
6.3.1.2. Content .....	28
6.3.1.3. Transfer Syntax .....	28
6.3.2. Body of Non-DICOM MIME Type Response .....	29
6.3.2.1. MIME Type .....	29
6.3.2.2. Content .....	29
6.4. WADO-WS Request/Response .....	29
6.4.1. WS - RetrievalImagingDocumentSet .....	29
6.4.1.1. Request .....	29
6.4.1.2. Response .....	30
6.4.1.2.1. Form of the Response .....	30
6.4.1.2.2. JPIP .....	32
6.4.2. WS - RetrieveRenderedImagingDocumentSet .....	32
6.4.2.1. Request .....	32
6.4.2.2. Response .....	33
6.4.3. WS - RetrievalImagingDocumentSetMetadataRequest .....	35
6.4.3.1. Request .....	35
6.4.3.2. Response .....	36
6.4.4. Error Codes .....	37
6.5. WADO-RS Request/Response .....	38
6.5.1. WADO-RS - RetrieveStudy .....	40
6.5.1.1. Request .....	40
6.5.1.2. Response .....	41
6.5.1.2.1. DICOM Response .....	42
6.5.1.2.2. Bulk Data Response .....	42
6.5.2. WADO-RS - RetrieveSeries .....	42
6.5.2.1. Request .....	43
6.5.2.2. Response .....	43
6.5.2.2.1. DICOM Response .....	43
6.5.2.2.2. Bulk Data Response .....	44
6.5.3. WADO-RS - RetrieveInstance .....	44
6.5.3.1. Request .....	44
6.5.3.2. Response .....	45
6.5.3.2.1. DICOM Response .....	45
6.5.3.2.2. Bulk Data Response .....	45
6.5.4. WADO-RS - RetrieveFrames .....	46

6.5.4.1. Request .....	46
6.5.4.2. Response .....	47
6.5.4.2.1. Pixel Data Response .....	47
6.5.5. WADO-RS - RetrieveBulkdata .....	47
6.5.5.1. Request .....	47
6.5.5.2. Response .....	48
6.5.5.2.1. Bulk Data Response .....	48
6.5.6. WADO-RS - RetrieveMetadata .....	49
6.5.6.1. Request .....	49
6.5.6.2. Response .....	50
6.5.6.2.1. XML Metadata Response .....	50
6.5.6.2.2. JSON Metadata Response .....	50
6.5.7. Error Codes .....	50
6.6. STOW-RS Request/Response .....	51
6.6.1. STOW-RS - Store Instances .....	51
6.6.1.1. Request .....	51
6.6.1.1.1. DICOM Request Message Body .....	52
6.6.1.1.2. XML Metadata and Bulk Data Request Message Body .....	52
6.6.1.1.3. JSON Metadata and Bulk Data Request Message Body .....	53
6.6.1.2. Action .....	54
6.6.1.3. Response .....	54
6.6.1.3.1. Response Status Line .....	54
6.6.1.3.2. Response Message Body .....	55
6.6.1.3.2.1. Store Instances Response Attribute Description .....	56
6.6.1.3.2.1.1. Warning Reason .....	56
6.6.1.3.2.1.2. Failure Reason .....	56
6.6.1.3.2.2. Response Message Body Example .....	57
6.7. QIDO-RS Request/Response .....	58
6.7.1. QIDO-RS - Search .....	59
6.7.1.1. Request .....	59
6.7.1.1.1. {attributeID} encoding rules .....	60
6.7.1.2. Response .....	61
6.7.1.2.1. Matching .....	61
6.7.1.2.1.1. Study Matching .....	62
6.7.1.2.1.2. Series Matching .....	62
6.7.1.2.1.3. Instance Matching .....	63
6.7.1.2.2. Query Result Attributes .....	63
6.7.1.2.2.1. Study Result Attributes .....	63
6.7.1.2.2.2. Series Result Attributes .....	64
6.7.1.2.2.3. Instance Result Attributes .....	65
6.7.1.2.3. Query Result Messages .....	65
6.7.1.2.3.1. XML Results .....	65
6.7.1.2.3.2. JSON Results .....	66
6.7.1.3. Status Codes .....	66
6.8. RS Capabilities Service .....	66
6.8.1. Retrieve Capabilities .....	66
6.8.1.1. Request Message .....	66
6.8.1.1.1. Method = OPTIONS .....	67
6.8.1.1.2. Header Fields .....	67
6.8.1.2. Response message .....	67
6.8.1.2.1. Resources .....	67
6.8.1.2.2. Methods .....	69
6.8.1.2.2.1. Retrieve Methods .....	69
6.8.1.2.2.2. Store Methods .....	70
6.8.1.2.2.3. Search Methods .....	71
6.8.1.2.2.4. Update Methods .....	72
6.8.1.2.2.5. Subscribe Methods .....	73
6.8.1.3. Status Codes .....	73
6.9. UPS-RS Worklist Service .....	74
6.9.1. CreateUPS .....	75

6.9.1.1. Request .....	75
6.9.1.1.1. Request Message .....	76
6.9.1.2. Behavior .....	76
6.9.1.3. Response .....	76
6.9.1.3.1. Response Status Line .....	76
6.9.1.3.2. Response Headers .....	77
6.9.1.3.3. Response Message Body .....	77
6.9.2. UpdateUPS .....	77
6.9.2.1. Request .....	77
6.9.2.1.1. Request Message .....	78
6.9.2.2. Behavior .....	78
6.9.2.3. Response .....	78
6.9.2.3.1. Response Status Line .....	78
6.9.2.3.2. Response Headers .....	79
6.9.2.3.3. Response Message Body .....	79
6.9.3. SearchForUPS .....	79
6.9.3.1. Request .....	79
6.9.3.2. Behavior .....	80
6.9.3.2.1. Matching .....	80
6.9.3.3. Response .....	80
6.9.3.3.1. Response Status Line .....	80
6.9.3.3.2. Query Result Attribute .....	81
6.9.3.3.3. Response Message .....	81
6.9.3.3.3.1. XML Response Message .....	81
6.9.3.3.3.2. JSON Response Message .....	81
6.9.4. RetrieveUPS .....	82
6.9.4.1. Request .....	82
6.9.4.2. Behavior .....	82
6.9.4.3. Response .....	82
6.9.4.3.1. Response Status Line .....	82
6.9.4.3.2. Response Message .....	83
6.9.4.3.2.1. XML Response Message .....	83
6.9.4.3.2.2. JSON Response Message .....	83
6.9.5. ChangeUPSSState .....	83
6.9.5.1. Request .....	83
6.9.5.1.1. Request Message .....	84
6.9.5.2. Behavior .....	84
6.9.5.3. Response .....	84
6.9.5.3.1. Response Status Line .....	84
6.9.5.3.2. Response Headers .....	85
6.9.5.3.3. Response Message Body .....	85
6.9.6. RequestUPSCancellation .....	85
6.9.6.1. Request .....	85
6.9.6.1.1. Request Message .....	86
6.9.6.2. Behavior .....	86
6.9.6.3. Response .....	86
6.9.6.3.1. Response Status Line .....	86
6.9.6.3.2. Response Headers .....	87
6.9.6.3.3. Response Message Body .....	87
6.9.7. CreateSubscription .....	87
6.9.7.1. Request .....	87
6.9.7.2. Behavior .....	88
6.9.7.3. Response .....	88
6.9.7.3.1. Response Status Line .....	88
6.9.7.3.2. Response Headers .....	89
6.9.7.3.3. Response Message Body .....	89
6.9.8. SuspendGlobalSubscription .....	89
6.9.8.1. Request .....	89
6.9.8.2. Behavior .....	89
6.9.8.3. Response .....	89

6.9.8.3.1. Response Status Line .....	89
6.9.8.2.2. Response Message Body .....	90
6.9.9. DeleteSubscription .....	90
6.9.9.1. Request .....	90
6.9.9.2. Behavior .....	90
6.9.9.3. Response .....	90
6.9.9.3.1. Response Status Line .....	90
6.9.9.3.2. Response Message Body .....	91
6.9.10. OpenEventChannel .....	91
6.9.10.1. Request .....	91
6.9.10.2. Behavior .....	91
6.9.10.3. Response .....	92
6.9.10.3.1. Response Status Line .....	92
6.9.10.3.2. Response Message Body .....	92
6.9.11. SendEventReport .....	92
6.9.11.1. Request .....	92
6.9.11.1.1. Request Message Body .....	92
6.9.11.2. Behavior .....	93
6.9.11.3. Response .....	93
7. Persistent Object Types .....	95
7.1. Single Frame Image Objects .....	95
7.1.1. Objects Accessed .....	95
7.1.2. MIME Type Constraints .....	95
7.2. Multi-frame and Video Image Objects .....	96
7.2.1. Objects Included .....	96
7.2.2. MIME Type Constraints .....	96
7.3. Text Objects .....	96
7.3.1. Objects Included .....	96
7.3.2. MIME Type Constraints .....	97
7.4. Other Objects .....	97
7.4.1. Objects Included .....	97
7.4.2. MIME Type Constraints .....	97
8. Parameters of the Request .....	99
8.1. Parameters Available for all DICOM Persistent Objects .....	99
8.1.1. Request Type .....	99
8.1.2. Unique Identifier of the Study .....	99
8.1.3. Unique Identifier of the Series .....	99
8.1.4. Unique Identifier of the Object .....	99
8.1.5. MIME Type of the Response .....	100
8.1.6. Charset of the Response .....	100
8.1.7. Anonymize Object .....	100
8.1.9. Retrieve Partial Information From Objects .....	101
8.2. Parameters for DICOM Image Persistent Objects .....	101
8.2.1. Annotation On The Object .....	101
8.2.2. Number of Pixel Rows .....	102
8.2.3. Number of Pixel Columns .....	102
8.2.4. Region of the Image .....	102
8.2.5. Window Center of the Image .....	103
8.2.6. Window Width of the Image .....	103
8.2.7. Frame Number .....	103
8.2.8. Image Quality .....	103
8.2.9. Unique Identifier of the Presentation Object .....	103
8.2.10. Unique Identifier of the Series Containing The Presentation Object .....	104
8.2.11. Transfer Syntax UID .....	104
A. URI Query Component Syntax (Normative) .....	105
B. Examples (Informative) .....	107
B.1. Retrieving a Simple DICOM Image in JPEG .....	107
B.2. Retrieving a DICOM SR in HTML .....	107
B.3. Retrieving a Region of A DICOM Image .....	107
B.4. Retrieving As A DICOM MIME Type .....	107

C. Applications (Informative) .....	109
D. IANA Mapping (Informative) .....	111
E. WADO WS Schemas and Examples .....	113
E.1. WADO WS XSD Schema (Informative) .....	113
E.2. WADO WS Request Example (Informative) .....	118
E.3. WADO WS Response Example .....	119
F. DICOM JSON Model .....	123
F.1. Introduction to JavaScript Object Notation (JSON) .....	123
F.2. DICOM JSON Model .....	123
F.2.1. Multiple Results Structure .....	123
F.2.1.1. Examples .....	123
F.2.1.1.1. Native DICOM Model .....	123
F.2.1.1.2. DICOM JSON Model .....	123
F.2.2. DICOM JSON Model Object Structure .....	124
F.2.3. DICOM JSON Value Representation .....	124
F.2.4. DICOM JSON Value Multiplicity .....	125
F.2.5. DICOM JSON Model Null Values .....	126
F.2.6. BulkDataURI .....	126
F.2.7. InlineBinary .....	126
F.3. Transformation with other DICOM Formats .....	126
F.3.1. Native DICOM Model XML .....	126
F.4. DICOM JSON Model Example .....	130
F.5. References .....	135
G. WADL JSON Representation .....	137
G.1. Introduction .....	137
G.2. XML Elements .....	137
G.2.1. Doc Elements .....	137
G.2.2. Unique Elements .....	137
G.2.3. Repeatable Elements .....	138



---

## List of Figures

6-1. Interaction Diagram .....	27
6.5-1. Mapping between IOD and HTTP message parts .....	39



## List of Tables

6.4-1. Error Codes .....	37
6.5-1. Media Type Mapping to Transfer Syntax .....	39
6.5-2. Error Codes .....	50
6.6.1-1. HTTP/1.1 Standard Response Code .....	54
6.6.1-2. Store Instances Response Module Attributes .....	55
6.7.1-1. QIDO-RS STUDY Search Query Keys .....	62
6.7.1-1a. QIDO-RS SERIES Search Query Keys .....	62
6.7.1-1b. QIDO-RS INSTANCE Search Query Keys .....	63
6.7.1-2. QIDO-RS STUDY Returned Attributes .....	63
6.7.1-2a. QIDO-RS SERIES Returned Attributes .....	64
6.7.1-2b. QIDO-RS INSTANCE Returned Attributes .....	65
6.7-1. QIDO-RS HTTP/1.1 Status Codes .....	66
6.8-1. Resources and Methods .....	67
6.8-2. Server Options HTTP/1.1 Status Codes .....	73
6.9-1. UPS Interface Mapping .....	75
6.9.1-1. Status Codes .....	76
6.9.2-1. Status Codes .....	78
6.9.3-1. Status Codes .....	80
6.9.4-1. Status Codes .....	83
6.9.5-1. Status Codes .....	84
6.9.6-1. Status Codes .....	86
6.9.7-2. Status Codes .....	88
6.9.8-1. Status Codes .....	90
6.9.7-1. Status Codes .....	91
6.9.10-1. Status Codes .....	92
D-1. IANA Mapping .....	111
F.2.3-1. DICOM VR to JSON Data Type Mapping .....	125
F.3.1-1. XML to JSON Mapping .....	127



# Notice and Disclaimer

The information in this publication was considered technically sound by the consensus of persons engaged in the development and approval of the document at the time it was developed. Consensus does not necessarily mean that there is unanimous agreement among every person participating in the development of this document.

NEMA standards and guideline publications, of which the document contained herein is one, are developed through a voluntary consensus standards development process. This process brings together volunteers and/or seeks out the views of persons who have an interest in the topic covered by this publication. While NEMA administers the process and establishes rules to promote fairness in the development of consensus, it does not write the document and it does not independently test, evaluate, or verify the accuracy or completeness of any information or the soundness of any judgments contained in its standards and guideline publications.

NEMA disclaims liability for any personal injury, property, or other damages of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, application, or reliance on this document. NEMA disclaims and makes no guaranty or warranty, expressed or implied, as to the accuracy or completeness of any information published herein, and disclaims and makes no warranty that the information in this document will fulfill any of your particular purposes or needs. NEMA does not undertake to guarantee the performance of any individual manufacturer or seller's products or services by virtue of this standard or guide.

In publishing and making this document available, NEMA is not undertaking to render professional or other services for or on behalf of any person or entity, nor is NEMA undertaking to perform any duty owed by any person or entity to someone else. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstances. Information and other standards on the topic covered by this publication may be available from other sources, which the user may wish to consult for additional views or information not covered by this publication.

NEMA has no power, nor does it undertake to police or enforce compliance with the contents of this document. NEMA does not certify, test, or inspect products, designs, or installations for safety or health purposes. Any certification or other statement of compliance with any health or safety-related information in this document shall not be attributable to NEMA and is solely the responsibility of the certifier or maker of the statement.



# Foreword

This DICOM Standard was developed according to the procedures of the DICOM Standards Committee.

The DICOM Standard is structured as a multi-part document using the guidelines established in ???.

PS3.1 should be used as the base reference for the current parts of this standard.



# 1 Scope

This standard specifies a web-based service for accessing and presenting DICOM (Digital Imaging and Communications in Medicine) persistent objects (e.g., images, medical imaging reports). This is intended for distribution of results and images to healthcare professionals. It provides a simple mechanism for accessing a DICOM persistent object, through HTTP/HTTPS protocol, using DICOM UIDs (Unique Identifiers). Data may be retrieved either in a presentation-ready form as specified by the requester (e.g., JPEG or GIF) or in a native DICOM format. It does not support facilities for web searching of DICOM images. This standard relates only to DICOM persistent objects (not to other DICOM objects or to non-DICOM objects). Access control beyond the security mechanisms generally available to web applications is outside the scope of this standard.



## 2 Conformance

Systems claiming conformance to this standard shall function in accordance with all its mandatory sections.



## 3 Normative References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this part of DICOM. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of DICOM are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ebRS ebXML Registry Service

HL7 CDA Health Level Seven, Clinical Document Architecture (CDA)

IETF RFC 822 Standard for ARPA Internet Text Messages <http://tools.ietf.org/html/rfc822>

IETF RFC 2045 and followings MIME Multipurpose Internet Mail Extension <http://tools.ietf.org/html/rfc2045>

IETF RFC 3240 Application/dicom MIME Sub-type Registration <http://tools.ietf.org/html/rfc3240>

IETF RFC 3986 Uniform Resource Identifiers (URI): Generic Syntax <http://tools.ietf.org/html/rfc3986>

IETF RFC 4627 The application/json Media Type for JavaScript Object Notation (JSON) <http://tools.ietf.org/html/rfc4627>

IETF RFC 6455 The WebSocket Protocol <http://tools.ietf.org/html/rfc6455>

IETF RFC 6570 URI Template <http://tools.ietf.org/html/rfc6570>

IETF RFC 7230 Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing <http://tools.ietf.org/html/rfc7230>

IETF RFC 7231 Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content <http://tools.ietf.org/html/rfc7231>

IETF RFC 7232 Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests <http://tools.ietf.org/html/rfc7232>

IETF RFC 7233 Hypertext Transfer Protocol (HTTP/1.1): Range Requests <http://tools.ietf.org/html/rfc7233>

IETF RFC 7234 Hypertext Transfer Protocol (HTTP/1.1): Caching <http://tools.ietf.org/html/rfc7234>

IETF RFC 7235 Hypertext Transfer Protocol (HTTP/1.1): Authentication <http://tools.ietf.org/html/rfc7235>

IETF RFC 7236 Initial Hypertext Transfer Protocol (HTTP) Authentication Scheme Registrations <http://tools.ietf.org/html/rfc7236>

IETF RFC 7237 Initial Hypertext Transfer Protocol (HTTP) Method Registrations <http://tools.ietf.org/html/rfc7237>

ISO/IEC 10918 JPEG Standard for digital compression and encoding of continuous-tone still images

IHE ITI TF-2x: Appendix V IHE IT Infrastructure Technical Framework, Volume 2x, Appendix V (Web Services for IHE Transactions)

SUBM-wadl-20090831 Web Application Description Language (WADL), W3C Member Submission 31 August 2009 <http://www.w3.org/Submission/wadl/>



## 4 Terms and Definitions

For the purposes of this part of DICOM, the following terms and definitions apply.

### 4.1 DICOM Persistent 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 Persistent Object is referred to as a Composite Service Object Pair (SOP) Instance.

### 4.2 Origin-Server

See IETF RFC-7230 Section 2.1 Client/Server Messaging.

### 4.3 User-Agent

See IETF RFC-7230 Section 2.1 Client/Server Messaging.

### 4.4 Web Client System

A system using Internet technologies (web, e-mail...) interested in retrieving DICOM Persistent Objects from a Web Enabled DICOM Server, through HTTP/HTTPS protocol.

### 4.5 Web Enabled DICOM Server

A system managing DICOM Persistent Objects and able to transmit them on request to the Web Client System.

### 4.6 Web Access to DICOM Persistent Objects

A service enabling the Web Client System to retrieve DICOM Persistent Objects managed by a Web Enabled DICOM Server, through HTTP/HTTPS protocol.



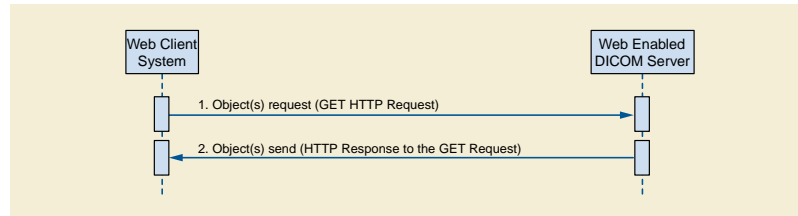
# 5 Symbols and Abbreviated Terms

<b>DICOM</b>	Digital Imaging and Communications in Medicine
<b>HL7</b>	Health Level Seven
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	HyperText Transfer Protocol
<b>HTTPS</b>	HyperText Transfer Protocol, secured
<b>IHE</b>	Integrating the Healthcare Enterprise
<b>MIME</b>	Multipurpose Internet Mail Extensions
<b>MTOM</b>	Message Transmission Optimization Mechanism
<b>QIDO-RS</b>	Query based on ID for DICOM Objects by RESTful Services
<b>REST</b>	Representational State Transfer
<b>RESTful</b>	A RESTful Web service is a Web service implemented using REST architecture and HTTP (see <a href="http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf">http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf</a> )
<b>SOAP</b>	Simple Object Access Protocol (SOAP12 for SOAP version 1.2)
<b>SOP</b>	Service Object Pair
<b>STOW-RS</b>	STore Over the Web by RESTful Services
<b>UID</b>	Unique (DICOM) Identifier
<b>UPS-RS</b>	Unified Procedure Step by RESTful Services
<b>URL/URI</b>	Uniform Resource Locator / Identifier
<b>WADL</b>	Web Application Description Language
<b>WADO-RS</b>	Web Access to DICOM Objects by RESTful Services
<b>WADO-URI</b>	Web Access to DICOM Objects by URI
<b>WADO-WS</b>	Web Access to DICOM Objects by Web Services (WS*)
<b>WS</b>	Web Services
<b>WSDL</b>	Web Services Description Language
<b>XML</b>	eXtensible Markup Language
<b>XOP</b>	XML-binary Optimized Packaging



# 6 Data Communication Requirements

## 6.1 Interaction



**Figure 6-1. Interaction Diagram**

The interaction shall be as shown in Figure 6-1.

Multiple communications modes are possible:

- URI based using HTTP Get: WADO-URI request
- Web Services (WS) using HTTP Post: WADO-WS, either:
  - a. DICOM Requester (Retrieve Imaging Document Set)
  - b. Rendered Requester (Retrieve Rendered Imaging Document Set)
  - c. Metadata Requester (Retrieve Imaging Document Set Metadata)
- RESTful Services (RS) using HTTP Get: WADO-RS, either:
  - a. DICOM Requester (Retrieve Study, Series, or Instance DICOM Objects)
  - b. Frame Pixel Data Requester (Retrieve Instance Frame Pixel Data)
  - c. Bulk Data Requester (Retrieve Study, Series, Instance Bulk Data)
  - d. Metadata Requester (Retrieve Study, Series, Instance Metadata)
- RESTful Services (RS) using HTTP Get: QIDO-RS:
  - a. Query Requester (Search for Study, Series or Instance DICOM Objects)
- RESTful Services (RS) using HTTP POST: STOW-RS, either:
  - a. DICOM Creator (Store Instances)
  - b. Metadata and Bulk Data Creator (Store Instances)
- RESTful Services (RS) using HTTP Options: RS Capabilities:
  - a. Provided information about the capabilities of a DICOM RESTful web service provider)

## 6.2 WADO-URI Request

The HTTP Request used shall use the GET method as defined in IETF RFC7230.

### 6.2.1 Parameters of the HTTP Request

The parameters of the <query> component of the Request-URI to be sent to the web Server through the HTTP GET method request shall be represented as defined in IETF RFC3986.

Note

1. Other components of the Request-URI depend on the configuration, e.g., location and script language of the Web Enabled DICOM Server.
2. The means by which the Web Client System obtains the value of the necessary parameters for web accessing of DICOM objects is out of the scope of the standard.

### 6.2.2 List of Media Types Supported in the Response

The "Accept" field of the GET method request shall specify the Media type(s) acceptable to the Web Client System. The(se) Media type(s) shall include at least the items of the list of MIME types specified in Section 7 of this standard devoted to the DICOM persistent object types.

Note

Typically the Accept field will be sent by a Web Client as "\*\*/\*". An optional parameter specifies the MIME type(s) preferred by the Web Client, as a subset of those specified in the "Accept" field.

### 6.2.3 List of Character Sets Supported in the Response

The "Accept-charset" field of the GET method request shall specify the character set of the object to be retrieved. If the "Accept-charset" field of the GET method is not present, or the Web Enabled DICOM Server does not support the specified character set, the character set of the response will be at the discretion of the Web Enabled DICOM Server.

Note Typically the user of a Web Client does not have control over the "Accept-charset" field. An optional parameter specifies the character set to be used in the returned object.

## 6.3 WADO-URI Response

The response shall be an HTTP Response Message as specified in IETF RFC7230.

Note

The content of the message-body varies according to the Media type as defined below.

### 6.3.1 Body of Single DICOM MIME Sub-type Part Response

#### 6.3.1.1 MIME Type

The MIME type shall be 'application/dicom', as specified in IETF RFC3240.

#### 6.3.1.2 Content

The body content shall be a "Part 10 File" that includes a meta-header as defined in PS3.10.

#### 6.3.1.3 Transfer Syntax

The returned DICOM object shall be encoded using one of the transfer syntaxes specified in the transfer syntax query parameter as defined in Section 8.2.11 below. By default, the transfer syntax shall be "Explicit VR Little Endian".

Note

This implies that retrieved images are sent uncompressed by default.

## 6.3.2 Body of Non-DICOM MIME Type Response

### 6.3.2.1 MIME Type

The MIME type shall be one on the MIME types defined in the contentType parameter, preferably the most desired by the Web Client, and shall be in any case compatible with the 'Accept' field of the GET method.

Note

The HTTP behavior is that an error (406 - Not Acceptable) is returned if the required content type cannot be served.

### 6.3.2.2 Content

The content shall be a single MIME part containing the object to be retrieved.

Note

Multiple objects in a response are not supported by this standard. The parameters select only a single object to retrieve. Most current Web Clients are able to retrieve single objects, within a "non multipart" MIME body, and are not able to support multipart/related or multipart/mixed responses.

## 6.4 WADO-WS Request/Response

The DICOM Web Service defines several action types. An implementation shall support at least one of these actions. The three action types are:

#### 1. RetrievalImagingDocumentSet

This action retrieves a set of DICOM instances and other objects. This action corresponds to the IHE XDS-I.b transaction RAD-69. The DICOM instances are formatted in accordance with PS3.10, and encapsulated in a Web Services response.

#### 2. RetrieveRenderedImagingDocumentSet

This action retrieves a set of DICOM instances that have been rendered into the requested format. For example, if rendering into JPEG was requested, these will be the JPEG renderings of the requested set of DICOM objects.

#### 3. RetrievalImagingDocumentSetMetadata

This action retrieves a set of DICOM instances presented as an Infoset with the bulk data removed. This service can retrieve either the full metadata, or a subset selected by XPath arguments. The XML encoding for the DICOM attributes is defined in PS3.19.

The Web Services actions shall be fully compliant with the Basic Profile of WS-I as defined in IHE IT Infrastructure Technical Framework Vol 2x Annex V. All <wsa:Action> elements shall have the mustUnderstand attribute set (mustUnderstand="1").

### 6.4.1 WS - RetrievalImagingDocumentSet

#### 6.4.1.1 Request

The specific Web Services parameters to be used for the Retrieve Imaging Document Set action shall be as follows, in the order that they would appear in the WSDL definition:

- The following types shall be imported (xsd:import) in the /definitions/types section:
  - namespace="urn:ihe:rad:xdsi-b:2009", schema="XDSI.b\_ImagingDocumentSource.xsd"
  - The baseline XDS.b schema (namespace="urn:ihe:iti:xds-b:2007", schema="XDS.b\_DocumentRepository.xsd")
  - The baseline DICOM WADO-WS schema (namespace="urn:dicom:wado:ws:2011", schema="dicom.wado.ws.2011.xsd")
- The /definitions/message/part/@element attribute of the Retrieve Imaging Document Set Request message shall be an "iherad:RetrievalImagingDocumentSetRequest" as defined below.

- The /definitions/message/part/@element attribute of the Retrieve Imaging Document Set Response message shall be an "ihe:RetrieveDocumentSetResponse" as defined below.
- The /definitions/portType/operation/input/@wsaw:Action attribute for the Retrieve Imaging Document Set Request message shall be "urn:ihe:rad:2009:RetrieveImagingDocumentSet".
- The /definitions/portType/operation/output/@wsaw:Action attribute for the Retrieve Imaging Document Set Response message shall be "urn:ihe:iti:2007:RetrieveDocumentSetResponse".
- The /definitions/binding/operation/soap12:operation/@soapAction attribute shall be "urn:ihe:rad:2009:RetrieveImagingDocumentSet".

The <iherad:RetrieveImagingDocumentSetRequest/> element for use with the Retrieve Imaging Document Set Request Message is defined as:

- One or more <iherad:StudyRequest/> elements each of which includes a "studyInstanceUID" attribute identifying the study associated with the DICOM images/ objects being retrieved. Each <iherad:StudyRequest/> element shall contain:
  - One or more <iherad:SeriesRequest/> elements each of which includes a "seriesInstanceUID" attribute identifying the series associated with the DICOM images/ objects being retrieved. Each <iherad:SeriesRequest/> element shall contain:
    - One or more <ihe:DocumentRequest/> elements, each one representing an individual document that the requestor wants to retrieve from the Web Server. Each <ihe:DocumentRequest/> element contains:
      - An optional <ihe:RepositoryUniqueid/> element that identifies the Web Server from which the document is to be retrieved. This value corresponds to XSDDocumentEntry.repositoryUniqueid. The RepositoryUniqueid is similar to a DICOM AETitle, but is a uniqueid assigned to the WADO-WS Web Server rather than a locally assigned string identifier. There will be a separate RepositoryUniqueid for each web service end point.
      - A required <ihe:DocumentUniqueid/> element that identifies the document within the source. For example, this value could be a SOP Instance UID obtained from a Key Object Selection (KOS) instance.
      - An optional <ihe:HomeCommunityId/> element. See the IHE Profiles for the definition and possible uses of this element.
      - An optional <wado:Anonymize/> element.
      - An optional <wado:FrameNumber/> element.
      - A required <iherad:TransferSyntaxUIDList/> element that contains a list of one or more <ihe:TransferSyntaxUID> elements. Each of the <iherad:TransferSyntaxUID> elements represent one of the transfer syntax encodings that the Imaging Document Consumer is capable of processing.

## 6.4.1.2 Response

A Web Server shall provide the document(s) indicated in the request. The Web Server shall return the document(s) or an error code when the document could not be returned. The pixel data shall be encoded using one of the DICOM transfer syntaxes included in the Retrieve Imaging Document Set Request Message. If the Imaging Document Source cannot encode the pixel data using any of the requested transfer syntaxes then an error status shall be returned.

### 6.4.1.2.1 Form of the Response

The <ihe:RetrieveDocumentResponse/> element for use with the Retrieve Imaging Document Set Response Message is defined as:

- A required /ihe:RetrieveDocumentSetResponse/rs:RegistryResponse element
- An optional sequence of <ihe:DocumentResponse/> elements containing
  - An optional <ihe:HomeCommunityId/> element. The value of this element shall be the same as the value of the /RetrieveImagingDocumentSetRequest/StudyRequest/SeriesRequest/DocumentRequest/HomeCommunityId element in the Retrieve Document Set Request Message. If the <ihe:HomeCommunityId/> element is not present in the Retrieve Document Set Request Message, this value shall not be present.

- An optional `<ihe:RepositoryUniqueId/>` that identifies the Imaging Document Source from which the document is to be retrieved. The value of this element shall be the same as the value of the `/RetrieveImagingDocumentSetRequest/StudyRequest/SeriesRequest/DocumentRequest/RepositoryUniqueId` element in the original Retrieve Imaging Document Set Request Message. This value corresponds to `XSDSDocumentEntry.repositoryUniqueId`.
- A required `<ihe:DocumentUniqueId/>` that identifies the document within the Imaging Document Source. The value of this element shall be the same as the value of the `/RetrieveImagingDocumentSetRequest/StudyRequest/SeriesRequest/DocumentRequest/DocumentUniqueId` element in the original Retrieve Imaging Document Set Request Message. This value corresponds to the SOP Instance UID in the Retrieve Document Request.
- A conditional `<wado:FrameNumber/>` that identifies the frame within the source document. It shall be present if and only if `<wado:FrameNumber/>` was in the request
- A required `<ihe:Document/>` element that contains the retrieved document as an XOP Infoset.
- A required `<ihe:mimeType/>` element that indicates the MIME type of the retrieved document.

The `/RetrieveDocumentSetResponse/rs:RegistryResponse/@status` attributes provides the overall status of the request: It shall contain one of the following values:

- `urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success`
- `urn:ihe:iti:2007:ResponseStatusType:PartialSuccess`
- `urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure`

See ITI TF-2a: 4.1.13 Error Reporting for the interpretation of these values.

For each document requested in a `/RetrieveImagingDocumentSetRequest/StudyRequest/SeriesRequest/DocumentRequest` element:

- If the document is successfully retrieved (without warning) then no `/RetrieveDocumentSetResponse/rs:RegistryResponse/rs:RegistryErrorList/ rs:RegistryError` element shall be present and a `/RetrieveDocumentSetResponse/DocumentResponse/Document` element shall be returned containing the document as base64binary encoded data.
- If a warning is reported when retrieving the document, then a `/RetrieveDocumentSetResponse/rs:RegistryResponse/rs:RegistryErrorList/ rs:RegistryError` element shall be returned with:
  - `@severity` is `urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Warning`
  - `@errorCode` is specified
  - `@codeContext` contains the warning message
  - `@location` contains the `DocumentUniqueId` of the document requested
  - The document shall be returned in an instance of `/RetrieveDocumentSetResponse/DocumentResponse/Document` as base64binary encoded data. The returned document and warning are correlated via the `DocumentUniqueId`.
- If an error is reported when retrieving a document, then a `/RetrieveDocumentSetResponse/rs:RegistryResponse/rs:RegistryErrorList/ rs:RegistryError` element shall be returned with:
  - `@severity` is `urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error`
  - `@errorCode` is specified
  - `@codeContext` contains the error message
  - `@location` contains the `DocumentUniqueId` of the document requested
  - No corresponding `RetrieveDocumentSetResponse/DocumentResponse` element shall be returned

The error conditions for failures and associated error codes are given below in section 6.4.4. These errors shall be detected and the associated `errorCode` returned if that error occurs. Additional errors defined in the ebRS standard, in ITI TF-2: 4.1.13 "Error Reporting", and defined by the implementer may be returned.

#### 6.4.1.2.2 JPIP

If the Web Client specifies a transfer syntax field of 1.2.840.10008.1.2.4.94 (DICOM JPIP Referenced Transfer Syntax) or 1.2.840.10008.1.2.4.95 (DICOM JPIP Referenced Deflate Transfer Syntax), and the Web Server supports the requested transfer syntax the following behavior is expected:

- If the DICOM Image Object(s) already have the same JPIP transfer syntax as the one indicated in the request, the Retrieve Imaging Document Set Response shall include the DICOM Image Objects unchanged.
- If the DICOM Image Object(s) have a transfer syntax that differs from that of the request, the Retrieve Imaging Document Set Response shall include the DICOM image with the transfer syntax changed to the requested transfer syntax. In addition, the pixel data Attribute (7FE0,0010 tag) will have been removed and replaced with a Pixel Data Provider URL (0028,7FE0 tag). The URL represents the JPIP request and will include the specific target information.
- Upon receipt of this Retrieve Imaging Document Set Response, the Imaging Document Consumer may request the pixel data from the pixel data provider using the supplied URL. Additional parameters required by the application may be appended to the URL when accessing the pixel data provider.
- For example, a JPIP request for a 200 by 200 pixel rendition of the entire image can be constructed from the Pixel Data Provider URL as follows:
  - Pixel Data Provider URL (0028,7FE0) = `https://server.xxx/jpipserver.cgi?target=imgxyz.jp2`,
  - URL Generated by the application = `https://server.xxx/jpipserver.cgi?target=imgxyz.jp2&fsiz=200,200`

### 6.4.2 WS - RetrieveRenderedImagingDocumentSet

#### 6.4.2.1 Request

The specific Web Services parameters to be used for the Retrieve Rendered Imaging Document Set action shall be as follows, in the order that they would appear in the WSDL definition:

- The following types shall be imported (xsd:import) in the /definitions/types section:
  - namespace="urn:ihe:rad:xdsi-b:2009", schema="XDSI.b\_ImagingDocumentSource.xsd"
  - The baseline XDS.b schema (namespace="urn:ihe:iti:xds-b:2007", schema="XDS.b\_DocumentRepository.xsd")
  - The baseline DICOM WADO-WS schema (namespace="urn:dicom:wado:ws:2011", schema="dicom.wado.ws.2011.xsd")
- The /definitions/message/part/@element attribute of the Retrieve Rendered Imaging Document Set Request message shall be a "wado:RetrieveRenderedImagingDocumentSetRequest" as defined below.
- The /definitions/message/part/@element attribute of the Retrieve Rendered Imaging Document Set Response message shall be a "wado:RetrieveRenderedImagingDocumentSetResponse" as defined below.
- The /definitions/portType/operation/input/@wsaw:Action attribute for the Retrieve Rendered Imaging Document Set Request message shall be "urn:dicom:ws:wado:2011:RetrieveRenderedImagingDocumentSet".
- The /definitions/portType/operation/output/@wsaw:Action attribute for the Retrieve Imaging Document Set Response message shall be "urn:dicom:ws:wado:2011:RetrieveRenderedImagingDocumentSetResponse".
- The /definitions/binding/operation/soap12:operation/@soapAction attribute shall be "urn:dicom:ws:wado:2011:RetrieveRenderedImagingDocumentSet".

The <wado:RetrieveRenderedImagingDocumentSetRequest/> element for use with the Retrieve Imaging Document Set Request Message is defined as:

- One or more <wado:StudyRequest/> elements each of which includes a "studyInstanceUID" attribute identifying the study associated with the DICOM images/ objects being retrieved. Each <iherad:StudyRequest/> element shall contain:

- One or more <wado:SeriesRequest/> elements each of which includes a "seriesInstanceUID" attribute identifying the series associated with the DICOM images/ objects being retrieved. Each <iherad:SeriesRequest/> element shall contain:
  - One or more <wado:RenderedDocumentRequest/> elements, each one representing an individual document that the requestor wants to retrieve from the Web Server. Each <wado:DocumentRequest/> element contains:
    - An optional <ihe:RepositoryUniqueid/> element that identifies the Web Server from which the document is to be retrieved. This value corresponds to XDSDocumentEntry.repositoryUniqueid. The RepositoryUniqueid is similar to a DICOM AETitle, but is a uniqueid assigned to the WADO-WS Web Server rather than a locally assigned string identifier. There will be a separate RepositoryUniqueid for each web service end point.
    - A required <ihe:DocumentUniqueid/> element that identifies the document within the source. This value corresponds to the SOP Instance UID referenced within the DICOM manifest.
    - An optional <ihe:HomeCommunityId/> element that corresponds to the home attribute of the Identifiable class in ebRIM.
    - An optional <wado:Annotation/> element.
    - An optional <wado:Rows/> element.
    - An optional <wado:Columns/> element.
    - An optional <wado:Region/> element.
    - An optional <wado:WindowCenter/> element.
    - An optional <wado:WindowWidth/> element.
    - An optional <wado:ImageQuality/> element.
    - An optional <wado:PresentationUID/> element.
    - An optional <wado:PresentationSeriesUID/> element.
    - An optional <wado:Anonymize/> element
    - An optional <wado:FrameNumber/> element.
    - A required <wado:ContentTypeList/> element that contains a list of one or more <wado:ContentType> elements.
    - An optional <wado:CharsetList/> element that contains a list of one or more <wado:Charset> elements.

### 6.4.2.2 Response

An Web Server shall render and then provide the document(s) indicated in the request. The Web Server shall return the rendered documents or an error code when the document could not be returned. The rendered forms shall be the subset specified, and in the format requested. If the Imaging Document Source cannot render the pixel data in that manner then an error status shall be returned.

The <wado:RetrieveRenderedImagingDocumentResponse/> element for use with the Retrieve Imaging Document Set Response Message, Retrieve Rendered Imaging Document Set Response Message and Retrieve Imaging Document Set Metadata Response Message is defined as:

- A required /ihe:RetrieveDocumentSetResponse/rs:RegistryResponse element
- An optional sequence of <wado:RenderedDocumentResponse/> elements containing:
  - A <ihe:HomeCommunityId/> element. The value of this element shall be the same as the value of the StudyRequest/SeriesRequest/DocumentRequest/HomeCommunityId element in the Request Message. If the <ihe:HomeCommunityId/> element is not present in the Request Message, this value shall not be present.
  - A required <ihe:RepositoryUniqueid/> that identifies the Imaging Document Source from which the document was retrieved. The value of this element shall be the same as the value of the StudyRequest/SeriesRequest/DocumentRequest/RepositoryUniqueid element in the original Request Message.

- A required `<wado:SourceDocumentUniqueid/>` that identifies the source document. The value of this element shall be the same as the value of the `StudyRequest/SeriesRequest/DocumentRequest/DocumentUniqueid` element in the original Request Message. This value identifies the source, and is not an ID for the resulting rendered document.
- A conditional `<wado:FrameNumber/>` that identifies the frame within the source document. It shall be present if and only if `<wado:FrameNumber/>` was in the request.
- A required `<wado:Annotation/>` element that contains the actual value used.
- A required `<wado:Rows/>` element that contains the actual value used.
- A required `<wado:Columns/>` element that contains the actual value used.
- A required `<wado:Region/>` element that contains the actual value used.
- A required `<wado:WindowCenter/>` element that contains the actual value used.
- A required `<wado:WindowWidth/>` element that contains the actual value used.
- A required `<wado:ImageQuality/>` element that contains the actual value used.
- A required `<wado:PresentationUID/>` element that contains the actual value used if a `PresentationUID` was used.
- A required `<wado:PresentationSeriesUID/>` element that contains the actual value used if a `PresentationSeriesUID` was used.
- An optional `<wado:Anonymize/>` element that shall be present if the rendered instance was anonymized.
- A required `<ihe:Document/>` element that contains the rendered document encoded as an XOP InfoSet.
- A required `<ihe:mimeType/>` element that indicates the MIME type of the retrieved document.

The `/RetrieveDocumentSetResponse/rs:RegistryResponse/@status` attributes provides the overall status of the request: It shall contain one of the following values:

- `urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success`
- `urn:ihe:iti:2007:ResponseStatusType:PartialSuccess`
- `urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure`

For each document requested in a `/RetrieveRenderedImagingDocumentSetRequest/StudyRequest/SeriesRequest/DocumentRequest` element:

- If the document is successfully rendered (without warning) then no `/RetrieveRenderedImagingDocumentSetResponse/rs:RegistryResponse/rs:RegistryErrorList/ rs:RegistryError` element shall be present and a `/RetrieveRenderedImagingDocumentSetResponse/DocumentResponse/Document` element shall be returned containing the rendered document as base64binary encoded data.
- If a warning is reported when retrieving the document, then a `/RetrieveRenderedImagingDocumentSetResponse/rs:RegistryResponse/rs:RegistryErrorList/ rs:RegistryError` element shall be returned with:
  - `@severity` is `urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Warning`
  - `@errorCode` is specified
  - `@codeContext` contains the warning message
  - `@location` contains the `DocumentUniqueid` of the document requested
- The rendered document shall be returned in an instance of `/RetrieveRenderedImagingDocumentSetResponse/DocumentResponse/Document` as base64binary encoded data. The returned document and warning are correlated via the `DocumentUniqueid`.
- If an error is reported when retrieving a document, then a `/RetrieveRenderedImagingDocumentSetResponse/rs:RegistryResponse/rs:RegistryErrorList/ rs:RegistryError` element shall be returned with:

- @severity is urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error
- @errorCode is specified
- @codeContext contains the error message
- @location contains the DocumentUniqueid of the document requested
- No corresponding RetrieveRenderedImagingDocumentSetResponse/DocumentResponse element shall be returned

The error conditions for failures and associated error codes are given below in section 6.4.4. These errors shall be detected and the associated errorCode returned if that error occurs. Additional errors defined in the ebRS standard, in ITI TF-2: 4.1.13 "Error Reporting", and defined by the implementer may be returned.

## 6.4.3 WS - RetrieveImagingDocumentSetMetadataRequest

### 6.4.3.1 Request

The specific Web Services parameters to be used for the Retrieve Imaging Document Set Metadata action shall be as follows, in the order that they would appear in the WSDL definition:

- The following types shall be imported (xsd:import) in the /definitions/types section:
  - namespace="urn:ihe:rad:xdsi-b:2009", schema="XDSI.b\_ImagingDocumentSource.xsd"
  - The baseline XDS.b schema (namespace="urn:ihe:iti:xds-b:2007", schema="XDS.b\_DocumentRepository.xsd")
  - The baseline DICOM WADO-WS schema (namespace="urn:dicom:wado:ws:2011", schema="dicom.wado.ws.2011.xsd")
- The /definitions/message/part/@element attribute of the Retrieve Imaging Document Information Set Request message shall be defined an "wado:RetrieveImagingDocumentSetInformationRequest" as defined below.
- The /definitions/message/part/@element attribute of the Retrieve Imaging Document Set Information Response message shall be defined an "wado:RetrieveImagingDocumentSetInformationResponse" as defined below.
- The /definitions/portType/operation/input/@wsaw:Action attribute for the Retrieve Imaging Document Set Information Request message shall be "urn:wado:2011:RetrieveImagingDocumentSetInformation".
- The /definitions/portType/operation/output/@wsaw:Action attribute for the Retrieve Imaging Document Set Information Response message shall be "urn:wado:2011:RetrieveImagingDocumentSetInformationResponse".
- The /definitions/binding/operation/soap12:operation/@soapAction attribute shall be "urn:wado:2011:RetrieveImagingDocumentSet-Information".

The <wado:RetrieveImagingDocumentSetInformationRequest/> element for use with the Retrieve Imaging Document Set Request Message is defined as:

- One or more <wado:StudyRequest/> elements each of which includes a "studyInstanceUID" attribute identifying the study associated with the DICOM images/objects being retrieved. Each <iherad:StudyRequest/> element shall contain:
  - One or more <wado:SeriesRequest/> elements each of which includes a "seriesInstanceUID" attribute identifying the series associated with the DICOM images/objects being retrieved. Each <iherad:SeriesRequest/> element shall contain:
    - One or more <wado:DocumentInformationRequest/> elements, each one representing an individual document that the requestor wants to retrieve from the Web Server. Each < wado:DocumentInformationRequest /> element contains:
      - An required <ihe:RepositoryUniqueid/> element that identifies the Web Server from which the document is to be retrieved. This value corresponds to XDSDocumentEntry.repositoryUniqueid. The RepositoryUniqueID is similar to a DICOM AE Title, but is a uniqueID assigned to the WADO-WS Web Server rather than a locally assigned string identifier. There will be a separate RepositoryUniqueID for each web service end point.
      - A required <ihe:DocumentUniqueid/> element that identifies the document within the source. For example, this value could be a SOP Instance UID obtained from a Key Object Selection (KOS) instance.

- An optional <ihe:HomeCommunityId/> element. See the IHE Profiles for the definition and possible uses of this element.
- An optional <wado:Anonymize/> element
- A required <wado:XPath/> that contains the text corresponding to the XPath "filter" applied to the Native DICOM Model transposition of the object, as defined in PS3.19.

#### Note

If the requested filter is "/", then all of the metadata is requested.

### 6.4.3.2 Response

A Web Server shall extract information from each document specified in a Document Set Information Request. This shall be done by the logical equivalent of:

1. convert the non-pixel data for each of the requested data into an XML encoded form
2. apply each of the wado:XPath elements to this XML encoded form
3. provide the XPath response as part of the Document Set Information Response.

See PS3.19 for details on conversion to XML encoded form.

The Web Server shall return the XPath results or an error code when the document could not be processed.

The <wado:RetrieveImagingDocumentSetInformationResponse/> element for use with the Retrieve Imaging Document Set Response Message is additionally defined as:

- A required /wado:RetrieveImagingDocumentSetInformationResponse/rs:RegistryResponse element
- An optional sequence of <wado:DocumentInformationResponse/> elements containing:
  - A <ihe:HomeCommunityId/> element. The value of this element shall be the same as the value of the StudyRequest/SeriesRequest/DocumentRequest/HomeCommunityId element in the Request Message. If the <ihe:HomeCommunityId/> element is not present in the Request Message, this value shall not be present.
  - A required <ihe:DocumentUniqueid/> that identifies the document within the Web Server. The value of this element shall be the same as the value of the StudyRequest/SeriesRequest/DocumentRequest/DocumentUniqueid element in the original Request Message. This value corresponds to the SOP Instance UID.
  - A conditional <wado:FrameNumber/> that identifies the frame within the source document. It shall be present if and only if <wado:FrameNumber/> was in the request.
  - One <wado:XPathResponseList/> containing:
    - A required <wado:XPathResponse> that contains the XPath results for each <wado:XPath/> elements, in the same order as in the request encoded as an XOP InfoSet. The response element shall be empty if there was no XPath match.

The /RetrieveImagingDocumentSetInformationResponse/rs:RegistryResponse/@status attributes provides the overall status of the request: It shall contain one of the following values:

- urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success
- urn:ihe:iti:2007:ResponseStatusType:PartialSuccess
- urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure

For each document requested in a /RetrieveImagingDocumentSetInformationRequest/StudyRequest/SeriesRequest/DocumentRequest element:

- If the document is successfully retrieved (without warning) then no `/RetrieveImagingDocumentSetInformationResponse/rs:RegistryResponse/rs:RegistryErrorList/ rs:RegistryError` element shall be present and a `/RetrieveImagingDocumentSetInformationResponse/DocumentResponse/Document` element shall be returned containing the document as base64binary encoded data.
- If a warning is reported when retrieving the document, then a `/RetrieveImagingDocumentSetInformationResponse/rs:RegistryResponse/rs:RegistryErrorList/ rs:RegistryError` element shall be returned with:
  - `@severity` is `urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Warning`
  - `@errorCode` is specified
  - `@codeContext` contains the warning message
  - `@location` contains the `DocumentUniqueld` of the document requested
  - The document shall be returned in an instance of `/RetrieveDocumentSetResponse/DocumentResponse/Document` as base64binary encoded data. The returned document and warning are correlated via the `DocumentUniqueld`.
- If an error is reported when retrieving a document, then a `/RetrieveImagingDocumentSetInformationResponse/rs:RegistryResponse/rs:RegistryErrorList/ rs:RegistryError` element shall be returned with:
  - `@severity` is `urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error`
  - `@errorCode` is specified
  - `@codeContext` contains the error message
  - `@location` contains the `DocumentUniqueld` of the document requested
  - No corresponding `RetrieveDocumentSetResponse/DocumentResponse` element shall be returned

The error conditions for failures and associated error codes are given below in section 6.4.4. These errors shall be detected and the associated `errorCode` returned if that error occurs. Additional errors defined in the ebRS standard, in ITI TF-2: 4.1.13 "Error Reporting", and defined by the implementer may be returned.

## 6.4.4 Error Codes

The following `errorCodes` are defined and shall be used to report any of the associated error and warning situations. Other `errorCodes` may be present for other error and warning situations.

**Table 6.4-1. Error Codes**

Error Code	Error Situation
urn:dicom:wado:0001	Unable to anonymize the requested instance(s).
urn:dicom:wado:0002	Web Server does not support anonymization.
urn:dicom:wado:0003	The requested instance(s) are not immediately available, but can be made available by manual request.
urn:dicom:wado:0004	Instance is no longer available, e.g., document retention rules have caused it to be removed or relocated.
urn:dicom:wado:0005	The requested instance(s) cannot be returned because the size or count exceeds resource limits.
urn:dicom:wado:0006	Web Server does not support the requested format or transfer syntax.
urn:dicom:wado:0007	The requested instance(s) cannot be provided in the requested format or transfer syntax.
urn:dicom:wado:0008	Single image format is not available for multi-frame images.
urn:dicom:wado:0009	Identifier does not match SOP Class (see PS3.7 C-MOVE)

Error Code	Error Situation
urn:dicom:wado:0010	Inconsistent identifiers, e.g., Study and Series are correct but Series is in a different Study (see PS3.7 C-MOVE)
urn:dicom:wado:0011	SOP Class not supported (see PS3.7 C-MOVE)
urn:dicom:wado:0012	Invalid parameter value in request (see PS3.7 C-MOVE)
urn:dicom:wado:0013	Unsupported parameter in request (see PS3.7 C-MOVE)
urn:dicom:wado:0014	Processing Failure (see PS3.7 C-MOVE)
urn:dicom:wado:0015	Study Instance UID not known
urn:dicom:wado:0016	Series Instance UID not known
urn:dicom:wado:0017	Document UID not known
urn:dicom:wado:0018	Out of range Frame number
urn:dicom:wado:0019	Presentation UID not known
urn:dicom:wado:0020	Presentation Series UID not known

## 6.5 WADO-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 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.

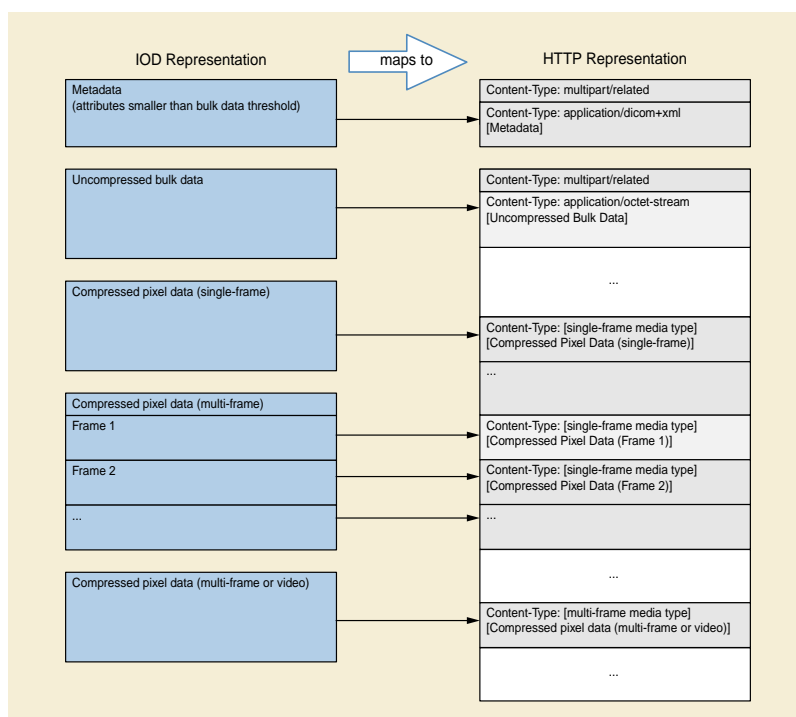
### 5. RetrieveBulkdata

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 study, series, or instance metadata with the bulk data removed.

DICOM objects returned shall be PS3.10 binary objects encoded in a requested Transfer Syntax (Explicit VR Little Endian by default) with one message part per DICOM Instance.



**Figure 6.5-1. Mapping between IOD and HTTP message parts**

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.
- Compressed pixel data may be encoded in one of three ways:
  - Single-frame pixel data encoded using a single-frame media type (one message part)
  - Multi-frame pixel data encoded using a single-frame media type (one frame per message part)
  - Multi-frame or video pixel data encoded using a multi-frame media type (multiple frames in one message part)

Compressed pixel data shall be encoded using the following Media Types. Media Types corresponding to several DICOM Transfer Syntax UUIDs require a transfer-syntax parameter, as shown in Table 6.5-1, to disambiguate the request.

**Note**

If the Transfer Syntax is not specified, then a reversible (lossless) encoding is used.

**Table 6.5-1. Media Type Mapping to Transfer Syntax**

DICOM Transfer Syntax UID	Media Type and Parameters
<b>Single-frame media types</b>	
1.2.840.10008.1.2.4.50	image/dicom+jpeg; transfer-syntax=1.2.840.10008.1.2.4.50
1.2.840.10008.1.2.4.51	image/dicom+jpeg; transfer-syntax=1.2.840.10008.1.2.4.51

DICOM Transfer Syntax UID	Media Type and Parameters
1.2.840.10008.1.2.4.57	image/dicom+jpeg; transfer-syntax=1.2.840.10008.1.2.4.57
1.2.840.10008.1.2.4.70	image/dicom+jpeg
1.2.840.10008.1.2.4.70	image/dicom+jpeg; transfer-syntax=1.2.840.10008.1.2.4.70
1.2.840.10008.1.2.5	image/dicom+rle
1.2.840.10008.1.2.5	image/dicom+rle; transfer-syntax=1.2.840.10008.1.2.5
1.2.840.10008.1.2.4.80	image/dicom+jpeg-ls
1.2.840.10008.1.2.4.80	image/dicom+jpeg-ls; transfer-syntax=1.2.840.10008.1.2.4.80
1.2.840.10008.1.2.4.81	image/dicom+jpeg-ls; transfer-syntax=1.2.840.10008.1.2.4.81
1.2.840.10008.1.2.4.90	image/dicom+jp2
1.2.840.10008.1.2.4.90	image/dicom+jp2; transfer-syntax=1.2.840.10008.1.2.4.90
1.2.840.10008.1.2.4.91	image/dicom+jp2; transfer-syntax=1.2.840.10008.1.2.4.91
1.2.840.10008.1.2.4.92	image/dicom+jpx
1.2.840.10008.1.2.4.92	image/dicom+jpx; transfer-syntax=1.2.840.10008.1.2.4.92
1.2.840.10008.1.2.4.93	image/dicom+jpx; transfer-syntax=1.2.840.10008.1.2.4.93
<b>Multi-frame media types</b>	
1.2.840.10008.1.2.4.92	image/dicom+jpx
1.2.840.10008.1.2.4.92	image/dicom+jpx; transfer-syntax=1.2.840.10008.1.2.4.92
1.2.840.10008.1.2.4.93	image/dicom+jpx; transfer-syntax=1.2.840.10008.1.2.4.93
1.2.840.10008.1.2.4.100	video/mpeg; transfer-syntax=1.2.840.10008.1.2.4.100
1.2.840.10008.1.2.4.101	video/mpeg; transfer-syntax=1.2.840.10008.1.2.4.101
1.2.840.10008.1.2.4.102	video/mp4; transfer-syntax=1.2.840.10008.1.2.4.102
1.2.840.10008.1.2.4.103	video/mp4; transfer-syntax=1.2.840.10008.1.2.4.103

**Note**

For the media type image/dicom+jp2 Transfer Syntaxes, 1.2.840.10008.1.2.4.90 and 1.2.840.10008.1.2.4.91, the image does not include the jp2 wrapper.

HTTP Request field Accept is used in the header lines by the client in a HTTP protocol transaction to indicate the data responses that are acceptable from the server. HTTP Response fields Content-Type and parameters are used in the header lines by the server in a HTTP protocol transaction to indicate the type and encoding of data returning to the client. All lines are RFC822 format headers. All HTTP header fields whose use is not defined by WADO-RS are presumed to have the meaning defined by the HTTP standard.

The server is required to support uncompressed bulk and pixel data (application/octet-stream) and must be able to deliver all bulk data in that form unless it is available only in a lossy-compressed format.

## 6.5.1 WADO-RS - 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.

### 6.5.1.1 Request

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

- Resource
  - {SERVICE}/studies/{StudyInstanceUID}, where

- {SERVICE} is the base URL for the service. This may be a combination of protocol (either http or https), host, port, and application.
- {StudyInstanceUID} is the study instance UID for a single study.
- Method
  - GET
- Headers
  - Accept - A comma-separated list of representation schemes, in preference order, which will be accepted by the service in the response to this request. The types allowed for this request header are as follows:
    - multipart/related; type=application/dicom; [transfer-syntax={TransferSyntaxUID}]
 

Specifies that the response can be DICOM Instances encoded in PS3.10 format. If *transfer-syntax* is not specified the server can freely choose which Transfer Syntax to use for each Instance.
    - multipart/related; type=application/octet-stream
 

Specifies that the response can be Little Endian uncompressed bulk data.
    - multipart/related; type={MediaType}
 

Specifies that the response can be pixel data encoded using a {MediaType} listed in Table 6.5-1 (including parameters).

#### Note

An example of a more complicated accept header with multiple transfer syntaxes:

User is interested in receiving JPEG2000 pixel data in lossless or compressed format but is willing to accept JPEG as well.

The Accept request would contain the following comma-separated parameters:

Accept: multipart/related=image/dicom+jpx; transfer-syntax=1.2.840.10008.1.2.4.92,, multipart/related=image/dicom+jpx; transfer-syntax=1.2.840.10008.1.2.4.93, multipart/related=image/dicom+jpeg

or alternatively, multiple Accept headers:

Accept: multipart/related=image/dicom+jpx; transfer-syntax=1.2.840.10008.1.2.4.92,

Accept: multipart/related=image/dicom+jpx; transfer-syntax=1.2.840.10008.1.2.4.93

Accept: multipart/related=application/dicom+jpeg

### 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 metadata for the Study.

The Server shall return the document(s), or an error code when the document(s) cannot be returned. If the server cannot convert all of the data to any of the requested media types/Transfer Syntaxes, then an error code shall be returned, either a "Not Acceptable" response if no data is returned or a "Partial Content" response if only some data is returned.

The client can compare the SOP Instance UIDs or bulk data URLs in the metadata and the message response to determine which bulk data elements have been returned.

All response formats have a content type of multipart/related with a message boundary separator. The response format depends on the Accept header specified in the request.

### 6.5.1.2.1 DICOM Response

- Content-Type:
  - multipart/related; type=application/dicom; boundary={MessageBoundary}
- The entire multipart response contains every instance for the specified Study that can be converted to one of the requested Transfer Syntaxes.
- Each item in the multipart response represents a DICOM SOP Instance with the following http headers:
  - Content-Type: application/dicom

### 6.5.1.2.2 Bulk Data Response

- Content-Type:
  - multipart/related; type=application/octet-stream; boundary={MessageBoundary}
  - multipart/related; type={MediaType}; boundary={MessageBoundary}
- The entire multipart response contains all bulk data for the specified Study that can be converted to one of the requested media types.
- Each item in the response is one of:
  - an uncompressed bulk data element encoded in Little Endian binary format with the following headers:
    - Content-Type: application/octet-stream
    - Content-Location: {BulkDataURL}
  - a compressed bulk data element from a SOP Instance in the Study encoded in a single-frame compression {MediaType} with the following headers:
    - Content-Type: {MediaType}
    - Content-Location: {BulkDataURL}
  - a compressed frame from a multi-frame SOP Instance in the Study encoded in a single-frame media type with the following headers:
    - Content-Type: {MediaType}
    - Content-Location: {BulkDataURL}/frames/{FrameNumber}

Note

Each frame will come in a separate part.

- a set of compressed frames from a SOP Instance in the Study encoded in a multi-frame media type with the following headers:
  - Content-Type: {MediaType}
  - Content-Location: {BulkDataURL}/frames/{FrameList}
  - {FrameList} is a list of frames separated by %2C (comma). It may be omitted if the message part includes all frames for the specified bulk pixel data object.

## 6.5.2 WADO-RS - RetrieveSeries

This action retrieves the set of DICOM instances associated with a given study and series UID. The response can be DICOM or bulk data depending on the "Accept" type, and is encapsulated in a multipart MIME response.

### 6.5.2.1 Request

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

- Resource
  - {SERVICE}/studies/{StudyInstanceUID}/series/{SeriesInstanceUID}, where
    - {SERVICE} is the base URL for the service. This may be a combination of protocol (either http or https), host, port, and application.
    - {StudyInstanceUID} is the study instance UID for a single study.
    - {SeriesInstanceUID} is the series instance UID for a single series.
- Method
  - GET
- Headers
  - Accept - A comma-separated list of representation schemes, in preference order, which will be accepted by the service in the response to this request. The types allowed for this request header are as follows:
    - multipart/related; type=application/dicom; [transfer-syntax={TransferSyntaxUID}]

Specifies that the response can be DICOM Instances encoded in PS3.10 format. If *transfer-syntax* is not specified the server can freely choose which Transfer Syntax to use for each Instance.
    - multipart/related; type=application/octet-stream;

Specifies that the response can be Little Endian uncompressed bulk data.
    - multipart/related; type={MediaType}

Specifies that the response can be pixel data encoded using a {MediaType} listed in Table 6.5-1 (including parameters).

### 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 corresponding metadata for the specified Study, Series, or Instance.

The Server shall return the document(s), or an error code when the document(s) cannot be returned. If the server cannot convert all of the data to any of the requested media types/Transfer Syntaxes, then an error code shall be returned, either a "Not Acceptable" response if no data is returned or a "Partial Content" response if only some data is returned.

The client can compare the SOP Instance UIDs or bulk data URLs in the metadata and the message response to determine which bulk data elements have been returned.

All response formats have a content type of multipart/related with a message boundary separator. The response format depends on the Accept header specified in the request.

#### 6.5.2.2.1 DICOM Response

- Content-Type:
  - multipart/related; type=application/dicom; boundary={MessageBoundary}
- The entire multipart response contains every instance for the specified Series that can be converted to one of the requested Transfer Syntaxes.
- Each item in the multipart response represents a DICOM SOP Instance with the following http headers:

- Content-Type: application/dicom

#### 6.5.2.2.2 Bulk Data Response

- Content-Type:
  - multipart/related; type= application/octet-stream; boundary={MessageBoundary}
  - multipart/related; type={MediaType}; boundary={MessageBoundary}
- The entire multipart response contains all bulk data for the specified Series that can be converted to one of the requested media types.
- Each item in the response is one of:
  - an uncompressed bulk data element encoded in Little Endian binary format with the following headers:
    - Content-Type: application/octet-stream
    - Content-Location: {BulkDataURL}
  - a compressed bulk data element from a SOP Instance in the Series encoded in a single-frame media type with the following headers:
    - Content-Type: {MediaType}
    - Content-Location: {BulkDataURL}
  - a compressed frame from a multi-frame SOP Instance in the Series encoded in a single-frame media type with the following headers:
    - Content-Type: {MediaType}
    - Content-Location: {BulkDataURL}/frames/{FrameNumber}
  - a set of compressed frames from a multi-frame SOP Instance in the Series encoded in a multi-frame media type with the following headers:
    - Content-Type: {MediaType}
    - Content-Location: {BulkDataURL}/{frames/{FrameList}}
    - {FrameList} is a list of frames separated by %2C (comma). It may be omitted if the message part includes all frames for the specified bulk pixel data object.

### 6.5.3 WADO-RS - RetrievalInstance

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.

#### 6.5.3.1 Request

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

- Resource
  - {SERVICE}/studies/{StudyInstanceUID}/series/{SeriesInstanceUID}/instances/{SOPInstanceUID}, where
    - {SERVICE} is the base URL for the service. This may be a combination of protocol (either http or https), host, port, and application.
    - {StudyInstanceUID} is the study instance UID for a single study.

- {SeriesInstanceUID} is the series instance UID for a single series.
- {SOPInstanceUID} is the SOP Instance UID for a single SOP Instance.
- Method
  - GET
- Headers
  - Accept - A comma-separated list of representation schemes, in preference order, which will be accepted by the service in the response to this request. The types allowed for this request header are as follows:
    - multipart/related; type=application/dicom; [transfer-syntax={TransferSyntaxUID}]
 

Specifies that the response can be DICOM Instances encoded in PS3.10 format. If *transfer-syntax* is not specified the server can freely choose which Transfer Syntax to use for each Instance.
    - multipart/related; type=application/octet-stream;
 

Specifies that the response can be Little Endian uncompressed bulk data.
    - multipart/related; type={MediaType}
 

Specifies that the response can be pixel data encoded using a {MediaType} listed in Table 6.5-1 (including parameters).

### 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 corresponding metadata for the specified Study, Series, or Instance.

The Server shall return the document(s), or an error code when the document(s) cannot be returned. If the server cannot convert all of the bulk data to any of the requested media types, then an error code shall be returned, either a "Not Acceptable" response if no data is returned or a "Partial Content" response if only some data is returned.

The client can compare the bulk data URLs in the metadata and the message response to determine which bulk data elements have been returned.

All response formats have a content type of multipart/related with a message boundary separator. The response format depends on the Accept header specified in the request.

#### 6.5.3.2.1 DICOM Response

- Content-Type:
  - multipart/related; type=application/dicom; boundary={MessageBoundary}
- The multipart response contains a single item representing the specified DICOM SOP Instance with the following http headers:
  - Content-Type: application/dicom

#### 6.5.3.2.2 Bulk Data Response

- Content-Type:
  - multipart/related; type=application/octet-stream; boundary={MessageBoundary}
  - multipart/related; type={MediaType}; boundary={MessageBoundary}
- The entire multipart response contains all bulk data for the specified Instance that can be converted to one of the requested media types.
- Each item in the response is one of:

- an uncompressed bulk data element encoded in Little Endian binary format with the following headers:
  - Content-Type: application/octet-stream
  - Content-Location: {BulkDataURL}
- a compressed bulk data element from a SOP Instance encoded in a single-frame media type with the following headers:
  - Content-Type: {MediaType}
  - Content-Location: {BulkDataURL}
- a compressed frame from a multi-frame SOP Instance encoded in a single-frame media type with the following headers:
  - Content-Type: {MediaType}
  - Content-Location: {BulkDataURL}/frames/{FrameNumber}
- a set of compressed frames from a multi-frame SOP Instance encoded in a multi-frame media type with the following headers:
  - Content-Type: {MediaType}
  - Content-Location: {BulkDataURL}/frames/{FrameList}
    - {FrameList} is a list of frames separated by %2C (comma). It may be omitted if the message part includes all frames for the specified bulk pixel data object.

## 6.5.4 WADO-RS - RetrieveFrames

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

### 6.5.4.1 Request

The specific Services resources to be used for the RetrieveFrames action shall be as follows:

- Resource
  - {SERVICE}/studies/{StudyInstanceUID}/series/{SeriesInstanceUID}/instances/{SOPInstanceUID}/frames/{FrameList}, where
    - {SERVICE} is the base URL for the service. This may be a combination of protocol (either http or https), host, port, and application.
    - {StudyInstanceUID} is the study instance UID for a single study.
    - {SeriesInstanceUID} is the series instance UID for a single series.
    - {SOPInstanceUID} is the SOP Instance UID for a single SOP Instance.
    - {FrameList} is a comma or %2C separated list of one or more non duplicate frame numbers. These may be in any order (e.g., ../frames/1,2,4,3).
- Method
  - GET
- Headers
  - Accept
    - multipart/related; type=application/octet-stream

Specifies that the response can be Little Endian uncompressed pixel data

- multipart/related; type={MediaType}

Specifies that the response can be pixel data encoded using a {MediaType} and {TransferSyntaxUID} listed in Table 6.5-1 (including parameters).

## 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 corresponding metadata for the specified Study, Series, or Instance.

The Server shall return the document(s) or an error code when the document(s) cannot be returned. If the server cannot encode the pixel data using any of the requested media types, then an error status shall be returned.

All response formats has a content type of multipart/related with a message boundary separator.

### 6.5.4.2.1 Pixel Data Response

- Content-Type:
  - multipart/related; type=application/octet-stream; boundary={MessageBoundary}
  - multipart/related; type={MediaType}; boundary={MessageBoundary}
- The entire multipart response contains all requested Frames for the specified Instance.
- Each item in the response is one of:
  - an uncompressed frame encoded in Little Endian binary format with the following headers:
    - Content-Type: application/octet-stream
    - Content-Location: {BulkDataURL}/{frames/{FrameNumber}}
  - a compressed frame encoded in a single-frame media type with the following headers:
    - Content-Type: {MediaType}
    - Content-Location: {BulkDataURL}/frames/{FrameNumber}
  - a set of compressed frames encoded in a multi-frame media type with the following headers:
    - Content-Type: {MediaType}
    - Content-Location: {BulkDataURL}/{frames/{FrameList}}
    - {FrameList} is a list of frames separated by %2C (comma). It may be omitted if the message part includes all frames for the specified bulk pixel data object.
- The frames will be returned in the order specified by the Frame List.

## 6.5.5 WADO-RS - RetrieveBulkdata

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

### 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 received in response to a WADO-RS RetrieveMetadataRequest.
- The server shall always return the same bulk data for a specified BulkData URL if the data is available.
- If the resource specified by the BulkData URL is not available, the server shall return:
  - 404 - Not Found, if the server expects to be able to return the resource again in the future
  - 410 - Gone, if the server does not expect the resource to be valid in the future
- The server determines the period of time a BulkData URL resource is available.
- Method
  - GET
- Headers
  - Accept
    - multipart/related; type=application/octet-stream  
Specifies that the response can be Little Endian uncompressed bulk data.
    - multipart/related; type={MediaType}  
Specifies that the response can be pixel data encoded using a {MediaType} listed in Table 6.5-1 (including parameters).
  - Range
    - See RFC 7230 Section 14.35. If omitted in the request the server shall return the entire bulk data object.

### 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 corresponding metadata for the specified Study, Series, or Instance.

The Server shall return the document(s) or an error code when the document(s) cannot be returned. If the server cannot encode the pixel data using any of the requested media types, then an error status shall be returned.

All response formats have a content type of multipart/related with a message boundary separator. The response format depends on the Accept header specified in the request.

#### 6.5.5.2.1 Bulk Data Response

- Content-Type:
  - multipart/related; type=application/octet-stream; boundary={MessageBoundary}
- The single item in the response is one of:
  - an uncompressed bulk data element encoded in Little Endian binary format with the following headers:
    - Content-Type: application/octet-stream
    - Content-Location: {BulkDataURL}
  - a compressed bulk data element from a SOP Instance encoded in a single-frame media type with the following headers:
    - Content-Type: {MediaType}
    - Content-Location: {BulkDataURL}

- If the Range header is specified in the request, the server shall return only the specified bytes of the bulk data object. See RFC 7230 Section 14.35.

### 6.5.6 WADO-RS - RetrieveMetadata

This action retrieves the DICOM instances presented as the study, series, or instance metadata with the bulk data removed. The response is metadata for the DICOM attributes.

The study, series, or instance metadata includes all attributes; however, a RESTful Service is permitted to replace the Value Field of an attribute with a BulkDataURL for attributes with Value Representations (VR) of FL, FD, IS, LT, OB, OD, OF, OW, SL, SS, ST, UL, UN, US, and UT. The client can use the BulkDataURL with the RetrieveBulkData action to retrieve the original Value Field of that attribute.

#### Note

1. The server is not required to replace any attribute with a BulkDataURL; this is intended to allow the server to provide clients with metadata of a reasonably small size by leaving out large data Value Fields.
2. Attributes with binary Value Fields are encoded as XML Base64 binary values.
3. Some DICOM instances, such as SR documents, may be entirely described in the metadata.

#### 6.5.6.1 Request

The specific Services resources to be used for the RetrieveMetadata action shall be as follows:

- Resources
  - {SERVICE}/studies/{StudyInstanceUID}/metadata
  - {SERVICE}/studies/{StudyInstanceUID}/series/{SeriesInstanceUID}/metadata
  - {SERVICE}/studies/{StudyInstanceUID}/series/{SeriesInstanceUID}/instances/{SOPInstanceUID}/metadata

where

- {SERVICE} is the base URL for the service. This may be a combination of protocol (either http or https), host, port, and application.
  - {StudyInstanceUID} is the study instance UID for a single study.
  - {SeriesInstanceUID} is the series instance UID for a single series.
  - {SOPInstanceUID} is the SOP Instance UID for a single SOP Instance.
- Method
    - GET
  - Headers
    - Accept
      - multipart/related; type=application/dicom+xml

Specifies that the response should be PS3.19 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 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 Annex F.

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

#### 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.

#### 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, Series, or Instance.
- 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 XML 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).

### 6.5.7 Error Codes

The following error codes are defined and shall be used to report any of the associated error and warning situations. Other error codes may be present for other error and warning situations.

**Table 6.5-2. Error Codes**

Client Error Code	Client Error Name	Error Situation
206	Partial Content	Accept type, Transfer Syntax or decompression method supported for some but not all requested content.
400	Bad Request	Malformed resource
404	Not Found	Specified resource does not exist

Client Error Code	Client Error Name	Error Situation
406	Not Acceptable	Accept type, Transfer Syntax or decompression method not supported
410	Gone	Specified resource was deleted
503	Busy	Service is unavailable

## 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 PS3.10 binary instances, or metadata and bulk data.

PS3.10 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 Mapping between IOD and HTTP message parts):

- All XML request messages shall be encoded as described in the Native DICOM Model defined in PS3.19 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)

Compressed pixel data shall be encoded using the Media Types as described in Table 6.5-1 WADO-RS Media Type Mapping to Transfer Syntax UID. Media Types corresponding to several DICOM Transfer Syntax UIDs may require a transfer-syntax parameter to disambiguate the request.

HTTP Request field Content-Type is used in the header lines by the client in an HTTP/1.1 transaction to indicate the type of data being sent to the Service. All lines are RFC822 or RFC7230 format headers. All HTTP header fields whose use is not defined by STOW-RS shall have the meaning defined by the HTTP standard.

The Service is required to support uncompressed bulk and pixel data (multipart/related; type= application/octet-stream).

### 6.6.1 STOW-RS - Store Instances

This action stores one or more DICOM instances associated with one or more study instance unique identifiers (SUID). The request message can be DICOM or metadata and bulk data depending on the "Content-Type", and is encapsulated in a multipart request body.

#### 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 PS3.10 binary instances. All STOW-RS providers must accept this Content-Type.
    - multipart/related; type=application/dicom+xml; boundary={messageBoundary}
 

Specifies that the post is PS3.19 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.

#### 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 9 in PS3.5 and B in PS3.5.

### 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

### 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 PS3.19 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 must be XML metadata.
- 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}
  - 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}
- 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:
  - 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.

### 6.6.1.2 Action

The Service may coerce or replace values of attributes such as Patient Name, ID, Accession Number, for example, during import of media from an external institution, reconciliation against a master patient index, or reconciliation against an imaging procedure order. The Service may correct, or replace incorrect values, such as Patient Name or ID, for example, when incorrect worklist item was chosen or operator input error occurs.

If any element is coerced or corrected, the Original Attribute Sequence (0400,0561) shall be included in the DICOM Object that is stored and may be included in the PS3.18 XML Store Instances Response Module in the response.

#### Note

For more information on populating the Original Attribute Sequence, see Section C.12.1 “SOP Common Module” in PS3.3.

### 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 a message body containing 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 PS3.19, or
- a DICOM JSON Model Object defined as defined in Annex F.

#### 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 Store Instances Response Module, 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 additional information regarding instance warnings or failures.

**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	<p>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> <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>
	415 - Unsupported Media Type	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).

Service Status	HTTP/1.1 Status Codes	STOW-RS Description
	503 - Busy	This indicates that the STOW-RS Service was unable to store any instances because it was out of resources.
Warning	202 - Accepted	This indicates that the STOW-RS Service stored some of the instances but warnings or failures exist for others.  Additional information regarding this error can be found in the XML response message body.
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 6.6.1.3.2.1. For example,

Warning: "A700: Out of memory"

### 6.6.1.3.2 Response Message Body

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

The 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 Store Instances Response Module in the response message body.

**Table 6.6.1-2. Store Instances Response Module Attributes**

Attribute Name	Tag	Type	Attribute Description
Retrieve URL	(0008,1190)	2	The URL where the Study is available for retrieval via a WADO-RS Retrieve Study service.  Note  The VR of this attribute has changed from UT to UR.
Failed SOP Sequence	(0008,1198)	1C	A sequence of Items where each Item references a single SOP Instance for which storage could not be provided.  Required if one or more SOP Instances failed to store.
<i>&gt;Table 10-11 "SOP Instance Reference Macro Attributes" in PS3.3</i>			
>Failure Reason	(0008,1197)	1	The reason that storage could not be provided for this SOP Instance.  See Section 6.6.1.3.2.1.2.
Referenced SOP Sequence	(0008,1199)	1C	A sequence of Items where each Item references a single SOP Instance that was successfully stored.  Required if one or more SOP Instances were successfully stored.
<i>&gt;Table 10-11 "SOP Instance Reference Macro Attributes" in PS3.3</i>			

Attribute Name	Tag	Type	Attribute Description
>Retrieve URL	(0008,1190)	2	The URL where the SOP Instance is available for retrieval via a WADO-RS service.  Note The VR of this attribute has changed from UT to UR.
>Warning Reason	(0008,1196)	1C	The reason that this SOP Instance was accepted with warnings.  Required if there was a warning for this SOP Instance.  See Section 6.6.1.3.2.1.1.
>Original Attributes Sequence	(0400,0561)	3	Sequence of Items containing all attributes that were removed or replaced by other values.  One or more Items are permitted in this sequence.
>>Attribute Modification DateTime	(0400,0562)	1	Date and time the attributes were removed and/or replaced.
>>Modifying System	(0400,0563)	1	Identification of the system that removed and/or replaced the attributes.
>>Reason for the Attribute Modification	(0400,0565)	1	Reason for the attribute modification. Defined terms are:  COERCE = Replace values of attributes such as Patient Name, ID, Accession Number, for example, during import of media from an external institution, or reconciliation against a master patient index.  CORRECT = Replace incorrect values, such as Patient Name or ID, for example, when incorrect worklist item was chosen or operator input error.
>>Modified Attributes Sequence	(0400,0550)	1	Sequence that contains all the Attributes, with their previous values, that were modified or removed from the main data set.  Only a single Item shall be included in this sequence.
>>Any Attribute from the main data set that was modified or removed; may include Sequence Attributes and their Items.			

#### 6.6.1.3.2.1 Store Instances Response Attribute Description

##### 6.6.1.3.2.1.1 Warning Reason

For the following semantics the associated value shall be used for the Warning Reason (0008,1196):

<b>B000 - Coercion of Data Elements</b>	The STOW-RS Service modified one or more data elements during storage of the instance. See Section 6.6.1.3.
<b>B006 - Elements Discarded</b>	The STOW-RS Service discarded some data elements during storage of the instance. See Section 6.6.1.3.
<b>B007 - Data Set does not match SOP Class</b>	The STOW-RS Service observed that the Data Set did not match the constraints of the SOP Class during storage of the instance.

Additional codes may be used for the Warning Reason (0008,1196) to address the semantics of other issues.

In the event that multiple codes may apply, the single most appropriate code shall be used.

##### 6.6.1.3.2.1.2 Failure Reason

For the following semantics the associated value shall be used for the Failure Reason (0008,1197). Implementation specific warning and error codes shall be defined in the conformance statement:

<b>A7xx - Refused out of Resources</b>	The STOW-RS Service did not store the instance because it was out of resources.
<b>A9xx - Error: Data Set does not match SOP Class</b>	The STOW-RS Service did not store the instance because the instance does not conform to its specified SOP Class.
<b>Cxxx - Error: Cannot understand</b>	The STOW-RS Service did not store the instance because it cannot understand certain Data Elements.
<b>C122 - Referenced Transfer Syntax not supported</b>	The STOW-RS Service did not store the instance because it does not support the requested Transfer Syntax for the instance.
<b>0110 - Processing failure</b>	The STOW-RS Service did not store the instance because of a general failure in processing the operation.
<b>0122 - Referenced SOP Class not supported</b>	The STOW-RS Service did not store the instance because it does not support the requested SOP Class.

Additional codes may be used for the Failure Reason (0008,1197) to address the semantics of other issues.

In the event that multiple codes may apply, the single most appropriate code shall be used.

#### 6.6.1.3.2.2 Response Message Body Example

The following is an example of a PS3.18 XML Store Instances Response Module in the response message body containing 2 failed SOP Instances, 1 successful SOP Instance, and 1 accepted SOP Instance with a warning:

```
<?xml version="1.0" encoding="utf-8"?>
<NativeDicomModel xmlns="http://dicom.nema.org/PS3.19/models/NativeDICOM"
xsi:schemaLocation="http://dicom.nema.org/PS3.19/models/NativeDICOM"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DicomAttribute tag="00081198" vr="SQ" keyword="FailedSOPSequence">
    <Item number="1">
      <DicomAttribute tag="00081150" vr="UI" keyword="ReferencedSOPClassUID">
        <Value number="1">1.2.840.10008.3.1.2.3.1</Value>
      </DicomAttribute>
      <DicomAttribute tag="00081155" vr="UI"
keyword="ReferencedSOPInstanceUID">
        <Value number="1">
          2.16.124.113543.6003.1011758472.49886.19426.2085542308</Value>
        </DicomAttribute>
      <DicomAttribute tag="00081197" vr="US" keyword="FailureReason">
        <Value number="1">290</Value>
      </DicomAttribute>
    </Item>
    <Item number="2">
      <DicomAttribute tag="00081150" vr="UI" keyword="ReferencedSOPClassUID">
        <Value number="1">1.2.840.10008.3.1.2.3.1</Value>
      </DicomAttribute>
      <DicomAttribute tag="00081155" vr="UI"
keyword="ReferencedSOPInstanceUID">
        <Value number="1">
          2.16.124.113543.6003.1011758472.49886.19426.2085542309</Value>
        </DicomAttribute>
      <DicomAttribute tag="00081197" vr="US" keyword="FailureReason">
        <Value number="1">290</Value>
      </DicomAttribute>
    </Item>
  </DicomAttribute>
  <DicomAttribute tag="00081199" vr="SQ" keyword="ReferencedSOPSequence">
```

```

<Item number="1">
  <DicomAttribute tag="00081150" vr="UI" keyword="ReferencedSOPClassUID">
    <Value number="1">1.2.840.10008.5.1.4.1.1.2</Value>
  </DicomAttribute>
  <DicomAttribute tag="00081155" vr="UI"
keyword="ReferencedSOPInstanceUID">
    <Value number="1">
      2.16.124.113543.6003.189642796.63084.16748.2599092903</Value>
    </DicomAttribute>
  <DicomAttribute tag="00081190" vr="UR" keyword="RetrieveURL">
    <Value number="1">
      https://wadors.hospital.com/studies/2.16.124.113543.6003.1154777499.30246.19789.3503430045/
      series/2.16.124.113543.6003.2588828330.45298.17418.2723805630/
      instances/2.16.124.113543.6003.189642796.63084.16748.2599092903</Value>
    </DicomAttribute>
  </Item>
<Item number="2">
  <DicomAttribute tag="00081150" vr="UI" keyword="ReferencedSOPClassUID">
    <Value number="1">1.2.840.10008.5.1.4.1.1.2</Value>
  </DicomAttribute>
  <DicomAttribute tag="00081155" vr="UI"
keyword="ReferencedSOPInstanceUID">
    <Value number="1">
      2.16.124.113543.6003.189642796.63084.16748.2599092905</Value>
    </DicomAttribute>
  <DicomAttribute tag="00081196" vr="US" keyword="WarningReason">
    <Value number="1">45056</Value>
  </DicomAttribute>
  <DicomAttribute tag="00081190" vr="UR" keyword="RetrieveURL">
    <Value number="1">
      https://wadors.hospital.com/studies/2.16.124.113543.6003.1154777499.30246.19789.3503430045/
      series/2.16.124.113543.6003.2588828330.45298.17418.2723805630/
      instances/2.16.124.113543.6003.189642796.63084.16748.2599092905</Value>
    </DicomAttribute>
  </Item>
</DicomAttribute>
<DicomAttribute tag="00081190" vr="UR" keyword="RetrieveURL">
  <Value number="1">
    https://wadors.hospital.com/studies/2.16.124.113543.6003.1154777499.30246.19789.3503430045</Value>
  </DicomAttribute>
</NativeDicomModel>

```

## 6.7 QIDO-RS Request/Response

DICOM QIDO-RS defines several action types. An implementation shall support the following action types:

### a. SearchForStudies

This action searches for DICOM Studies that match specified search parameters and returns a list of matching studies and the requested attributes for each study.

### b. SearchForSeries

This action searches for DICOM Series that match specified search parameters and returns a list of matching series and the requested attributes for each series.

### c. SearchForInstances

This action searches for DICOM Instances that match specified search parameters and returns a list of matching instances and the requested attributes for each instance.

## 6.7.1 QIDO-RS - Search

### 6.7.1.1 Request

The specific resources to be used for the search actions shall be as follows:

- Resource
  - SearchForStudies
    - {+SERVICE}/studies{?query\*,fuzzymatching,limit,offset}
  - SearchForSeries
    - {+SERVICE}/studies/{StudyInstanceUID}/series{?query\*,fuzzymatching,limit,offset}
    - {+SERVICE}/series{?query\*,fuzzymatching,limit,offset}
  - SearchForInstances
    - {+SERVICE}/studies/{StudyInstanceUID}/series/{SeriesInstanceUID}/instances{?query\*,fuzzymatching,limit,offset}
    - {+SERVICE}/studies/{StudyInstanceUID}/instances{?query\*,fuzzymatching,limit,offset}
    - {+SERVICE}/instances{?query\*,fuzzymatching,limit,offset}

where

- {+SERVICE} is the base URL for the QIDO RESTful service. This may be a combination of protocol (http or https), authority, and path.
- {StudyInstanceUID} is the unique Study Instance UID for a single study.
- {SeriesInstanceUID} is the unique Series Instance UID for a single series.
- Method
  - GET
- Headers
  - Accept - The Media Type of the query results. The types allowed for this request header are:
    - multipart/related; type=application/dicom+xml (default)  
Specifies that the results should be DICOM PS3.19 XML (one part per result)
    - application/json  
Specifies that the results should be DICOM JSON

A QIDO-RS provider shall support both Accept header values
  - Cache-control: no-cache (recommended)  
If included, specifies that search results returned should be current and not cached.
  - {query}
    - {attributeID}={value}  
0-n / {attributeID}={value} pairs allowed
    - includefield={attributeID} | all

0-n includefield / {attributeID} pairs allowed, where "all" indicates that all available attributes should be included for each response.

Each {attributeID} must refer to one of:

- Patient IE attributes
- Study IE attributes
- Series IE attributes (SearchForSeries or SearchForInstances requests only)
- Composite Instance IE attributes (SearchForInstances requests only)
- Additional Query/Retrieve Attributes (Section C.3.4 in PS3.4)
- Timezone Offset From UTC (0008,0201)

See Section 6.7.1.1.1 for {attributeID} and {value} encoding rules

- fuzzymatching=true | false
- limit={maximumResults}
- offset={skippedResults}

#### **6.7.1.1.1 {attributeID} encoding rules**

Each {attributeID} query key shall be unique unless the associated DICOM Attribute allows UID List matching (see Section C.2.2.2.2 in PS3.4), in which case each {value} will be interpreted to be an element of the UID List.

The acceptable values for {value} are determined by the types of matching allowed by C-FIND for its associated {attributeID} (see Section C.2.2.2 in PS3.4). All characters in {value} that are disallowed for URIs shall be percent-encoded. See IETF RFC 3986 for details.

If an {attributeID} is passed as the value of an "includefield" query key this is equivalent to C-FIND Universal matching for the specified attribute (see Section C.2.2.2.3 in PS3.4).

{attributeID} can be one of the following:

- {dicomTag}
- {dicomKeyword}
- {dicomTag}.{attributeID}, where {attributeID} is an element of the sequence specified by {dicomTag}
- {dicomKeyword}.{attributeID}, where {attributeID} is an element of the sequence specified by {dicomKeyword}

{dicomTag} is the eight character hexadecimal string corresponding to the Tag of a DICOM Attribute (see 6 in PS3.6).

{dicomKeyword} is the Keyword of a DICOM Attribute (see 6 in PS3.6).

#### **Note**

Examples of valid values for {attributeID}:

- 0020000D
- StudyInstanceUID
- 00101002.00100020
- OtherPatientIDsSequence.PatientID
- 00101002.00100024.00400032

- OtherPatientIDsSequence.IssuerOfPatientIDQualifiersSequence.UniversalEntityID

Note

Examples of valid QIDO-RS URLs:

- `http://dicomrs/studies?PatientID=11235813`
- `http://dicomrs/studies?PatientID=11235813&StudyDate=20130509`
- `http://dicomrs/studies?00100010=SMITH*&00101002.00100020=11235813&limit=25`
- `http://dicomrs/studies?00100010=SMITH*&OtherPatientIDsSequence.00100020=11235813`
- `http://dicomrs/studies?PatientID=11235813&includefield=00081048&includefield=00081049&includefield=00081060`
- `http://dicomrs/studies?PatientID=11235813&StudyDate=20130509-20130510`
- `http://dicomrs/studies?StudyInstanceUID=1.2.392.200036.9116.2.2.2.2162893313.1029997326.94587%2c1.2.392.200036.9116.2.2.2.2162893313.1029997326.94583`

### 6.7.1.2 Response

The Server shall perform the query indicated in the request. The Server shall return the query results or, when the query cannot be performed, an error code.

If the limit query key is not specified or its value exceeds the total number of matching results then {maximumResults} is the lesser of the number of matching results and the maximum number of results supported by the Server.

If the offset query key is not specified or its value is less than zero then {skippedResults} is zero.

The first result returned shall be result number ({skippedResults} + 1). The last result returned shall be result number ({skippedResults} + {maximumResults}). If ({skippedResults} + 1) exceeds {maximumResults} then no results are returned.

If the number of results exceeds the maximum supported by the server, the server shall return the maximum supported results and the response shall include the following HTTP/1.1 Warning header (see RFC 7230 Section 14.46):

Warning: 299 (SERVICE): "The number of results exceeded the maximum supported by the server. Additional results can be requested.

Note

The client can request additional results by specifying a value for the "offset" query key.

The server shall be idempotent so that if the list of results is the same, the response to a request with a specific set of parameters shall always be the same, including order. If the complete list of results is different for subsequent transactions the responses may be different. In a situation where results are changing due to changes in the server contents, queries using the limit and offset may be inconsistent.

The response format depends on the Accept header specified in the request.

#### 6.7.1.2.1 Matching

The matching semantics for each attribute are determined by the types of matching allowed by C-FIND (see Section C.2.2.2 in PS3.4).

Matching results shall be generated according to the Hierarchical Search Method described in Section C.4.1.3.1.1 in PS3.4.

Combined Datetime matching shall be performed (see Section C.2.2.2.5 in PS3.4).

Note

If a QIDO-RS provider is acting as a proxy for a C-FIND SCP that does not support combined Datetime matching the QIDO-RS provider will need to perform a C-FIND request using Date only and filter results outside the time range before returning a QIDO-RS response

If the TimezoneOffsetFromUTC / 00080201 query key is included in the request, dates and times in the request are to be interpreted in the specified time zone.

If the "fuzzymatching=true" query key/value is included in the request and it is supported then additional fuzzy semantic matching of person names shall be performed in the manner specified in the DICOM Conformance Statement for the service provider.

If the "fuzzymatching=true" query key/value is included in the request and it is not supported, the response shall include the following HTTP/1.1 Warning header (see RFC 7230 Section 14.46):

Warning: 299 {SERVICE}: "The fuzzymatching parameter is not supported. Only literal matching has been performed."

where {SERVICE} is the base URL for the QIDO-RS provider. This may be a combination of scheme (http or https), host, port, and application.

#### Note

The Warning header is separate from the Status Line and does not affect the returned Status Code.

#### 6.7.1.2.1.1 Study Matching

Providers of the SearchForStudies service shall support the search query keys described in Table 6.7.1-1:

**Table 6.7.1-1. QIDO-RS STUDY Search Query Keys**

Key Word	Tag
StudyDate	00080020
StudyTime	00080030
AccessionNumber	00080050
ModalitiesInStudy	00080061
ReferringPhysicianName	00080090
PatientName	00100010
PatientID	00100020
StudyInstanceUID	0020000D
StudyID	00200010

#### 6.7.1.2.1.2 Series Matching

Providers of the SearchForSeries service shall support the search query keys described in Table 6.7.1-1a:

**Table 6.7.1-1a. QIDO-RS SERIES Search Query Keys**

Key Word	Tag
Modality	00080060
SeriesInstanceUID	0020000E
SeriesNumber	00200011
PerformedProcedureStepStartDate	00400244
PerformedProcedureStepStartTime	00400245
RequestAttributeSequence	00400275
>ScheduledProcedureStepID	00400009
>RequestedProcedureID	00401001

If {StudyInstanceUID} is not specified in the URL and this form of Relational Query is supported, all Study-level attributes specified in Table 6.7.1-1 shall also be supported.

### 6.7.1.2.1.3 Instance Matching

Providers of the SearchForInstances service shall support the search query keys described in Table 6.7.1-1b:

**Table 6.7.1-1b. QIDO-RS INSTANCE Search Query Keys**

Key Word	Tag
SOPClassUID	00080016
SOPInstanceUID	00080018
InstanceNumber	00200013

If {StudyInstanceUID} is not specified in the URL and this form of Relational Query is supported, all Study-level attributes specified in Table 6.7.1-1 shall also be supported.

If {SeriesInstanceUID} is not specified in the URL and this form of Relational Query is supported, all Series-level attributes specified in Table 6.7.1-1a shall also be supported.

### 6.7.1.2.2 Query Result Attributes

#### 6.7.1.2.2.1 Study Result Attributes

For each matching Study, the QIDO-RS provider shall return all attributes in accordance with Table 6.7.1-2:

**Table 6.7.1-2. QIDO-RS STUDY Returned Attributes**

Attribute Name	Tag	Notes
Specific Character Set	(0008,0005)	If necessary for encoding any returned attributes
Study Date	(0008,0020)	
Study Time	(0008,0030)	
Accession Number	(0008,0050)	
Instance Availability	(0008,0056)	
Modalities in Study	(0008,0061)	
Referring Physician's Name	(0008,0090)	
Timezone Offset From UTC	(0008,0201)	May be absent if no value is available
Retrieve URL	(0008,1190)	Shall be empty if the resource cannot be retrieved via WADO-RS  Note  The VR of this attribute has changed from UT to UR.
Patient's Name	(0010,0010)	
Patient ID	(0010,0020)	
Patient's Birth Date	(0010,0030)	
Patient's Sex	(0010,0040)	
Study Instance UID	(0020,000D)	
Study ID	(0020,0010)	
Number of Study Related Series	(0020,1206)	
Number of Study Related Instances	(0020,1208)	
All other Study Level DICOM Attributes passed as {attributeID} query keys that are supported by the service provider as matching or return attributes		

Attribute Name	Tag	Notes
All other Study Level DICOM Attributes passed as "includefield" query values that are supported by the service provider as return attributes		
All available Study Level DICOM Attributes if the "includefield" query key is included with a value of "all"		

Series Level and Instance Level attributes passed as "includefield" query values shall not be returned.

**Note**

The above list is consistent with those required for IHE RAD-14 (see [http://www.ihe.net/Technical\\_Framework/upload/IHE\\_RAD\\_TF\\_Vol2.pdf](http://www.ihe.net/Technical_Framework/upload/IHE_RAD_TF_Vol2.pdf) Table 4.14-1).

#### 6.7.1.2.2.2 Series Result Attributes

For each matching Series, the QIDO-RS provider shall return all attributes listed in Table 6.7.1-2a:

**Table 6.7.1-2a. QIDO-RS SERIES Returned Attributes**

Attribute Name	Tag	Notes
Specific Character Set	(0008,0005)	If necessary for encoding any returned attributes
Modality	(0008,0056)	
Timezone Offset From UTC	(0008,0201)	May be absent if no value is available
Series Description	(0008,103E)	May be absent if no value is available
Retrieve URL	(0008,1190)	Shall be empty if the resource cannot be retrieved via WADO-RS  Note  The VR of this attribute has changed from UT to UR.
Series Instance UID	(0020,000E)	
Series Number	(0020,0011)	
Number of Series Related Instances	(0020,1209)	
Performed Procedure Step Start Date	(0040,0244)	May be absent if no value is available
Performed Procedure Step Start Time	(0040,0245)	May be absent if no value is available
Request Attribute Sequence	(0040,0275)	May be absent if no value is available
>Scheduled Procedure Step ID	(0040,0009)	
>Requested Procedure ID	(0040,1001)	
All other Series Level DICOM Attributes passed as {attributeID} query keys that are supported by the service provider as matching or return attributes		
All other Study or Series Level DICOM Attributes passed as "includefield" query values that are supported by the service provider as return attributes		
All available Instance Level DICOM Attributes if the "includefield" query key is included with a value of "all"		
If {StudyInstanceUID} is not specified, all Study-level attributes specified in Table 6.7.1-2		

Instance Level attributes passed as "includefield" query values shall not be returned.

**Note**

The above list is consistent with the attributes required for IHE RAD-14 (see [http://www.ihe.net/Technical\\_Framework/upload/IHE\\_RAD\\_TF\\_Vol2.pdf](http://www.ihe.net/Technical_Framework/upload/IHE_RAD_TF_Vol2.pdf) Table 4.14-1).

### 6.7.1.2.2.3 Instance Result Attributes

For each matching instance, the QIDO-RS provider shall return all attributes listed in Table 6.7.1-2b:

**Table 6.7.1-2b. QIDO-RS INSTANCE Returned Attributes**

Attribute Name	Tag	Notes
Specific Character Set	(0008,0005)	If necessary for encoding any returned attributes
SOP Class UID	(0008,0016)	
SOP Instance UID	(0008,0018)	
Instance Availability	(0008,0056)	
Timezone Offset From UTC	(0008,0201)	May be absent if no value is available
Retrieve URL	(0008,1190)	Shall be empty if the resource cannot be retrieved via WADO-RS  Note The VR of this attribute has changed from UT to UR.
Instance Number	(0020,0013)	
Rows	(0028,0010)	Only present for Image Instances
Columns	(0028,0011)	Only present for Image Instances
Bits Allocated	(0028,0100)	Only present for Image Instances
Number of Frames	(0028,0008)	Only present for Multi-frame image instances
All other Instance Level DICOM Attributes passed as {attributeID} query keys that are supported by the service provider as matching or return attributes		
All other Study, Series or Instance Level DICOM Attributes passed as "includefield" query values that are supported by the service provider as return attributes		
All available Instance Level DICOM Attributes if the "includefield" query key is included with a value of "all"		
If {StudyInstanceUID} is not specified, all Study-level attributes specified in Table 6.7.1-2		
If {SeriesInstanceUID} is not specified, all Series-level attributes specified in Table 6.7.1-2a		

#### Note

The above list is consistent with the attributes required for IHE RAD-14 (see [http://www.ihe.net/Technical\\_Framework/upload/IHE\\_RAD\\_TF\\_Vol2.pdf](http://www.ihe.net/Technical_Framework/upload/IHE_RAD_TF_Vol2.pdf) Table 4.14-1 and Table 4.14-2).

### 6.7.1.2.3 Query Result Messages

The server shall support returning query results as:

- XML Results
- JSON Results

The result format used shall depend on the Accept header of the request.

#### 6.7.1.2.3.1 XML Results

- Content-Type: multipart/related; type=application/dicom+xml
  - The response is a multipart message body where each part is a DICOM PS3.19 XML NativeDicomModel element containing the attributes for one matching Study, Series or Instance (see Section A.1 in PS3.19).

- The provider of the QIDO service may use a BulkData reference at its discretion (see Table A.1.5-2 in PS3.19 and Section 6.5.6). For example, this might be done to avoid encoding a large DICOM Value Field, such as an image thumbnail.
- If there are no matching results, the message body will be empty.
- Each item in the multipart response will contain the following HTTP/1.1 headers:
  - Content-Type: application/dicom+xml

#### 6.7.1.2.3.2 JSON Results

- Content-Type: application/json
- The response is a DICOM JSON message containing a DICOM JSON property for each matching study, series or instance containing sub-properties describing the matching attributes for each study, series or instance (see Section F.2).
- The provider of the QIDO service may use a BulkDataURI reference at its discretion (see Section F.2.6). For example, this might be done to avoid encoding a large DICOM Value Field, such as an image thumbnail.
- If there are no matching results, the JSON message is empty.

### 6.7.1.3 Status Codes

Table 6.7-1 lists the HTTP/1.1 status codes that shall be used to report any of the associated error and warning situations. Other error codes may be present for other error and warning situations.

**Table 6.7-1. QIDO-RS HTTP/1.1 Status Codes**

Code	Name	Description
Success		
200	OK	The query completed and any matching results are returned in the message body.
Failure		
400	Bad Request	The QIDO-RS Provider was unable to perform the query because the Service Provider cannot understand the query component.
401	Unauthorized	The QIDO-RS Provider refused to perform the query because the client is not authenticated.
403	Forbidden	The QIDO-RS Provider understood the request, but is refusing to perform the query (e.g., an authenticated user with insufficient privileges).
413	Request entity too large	The query was too broad and a narrower query or paging should be requested. The use of this status code should be documented in the conformance statement.
503	Busy	Service is unavailable.

## 6.8 RS Capabilities Service

DICOM RS Capabilities Service defines a single transaction type which shall be supported by all implementations

### a. RetrieveCapabilities

This transaction retrieves the parameters for services supported by the server.

### 6.8.1 Retrieve Capabilities

#### 6.8.1.1 Request Message

The Retrieve Server Options transaction can be requested for the following resources:

- {+SERVICE}/{InformationEntity\*}
  - where {+SERVICE} is the base URL for the service. This may be a combination of protocol (either http or https), host, port, and application.
  - where {InformationEntity} is the path to a defined DICOM RESTful service resource, such as:
    - WADO-RS (see 6.5.1, 6.5.2, 6.5.3, 6.5.4, 6.5.5, 6.5.6)
    - STOW-RS (see 6.6.1)
    - QIDO-RS (see 6.7.1)
    - UPS-RS (see 6.9.1, 6.9.2, 6.9.3, 6.9.4, 6.9.5, 6.9.6, 6.9.7, 6.9.8, 6.9.9)

#### 6.8.1.1.1 Method = OPTIONS

The Retrieve Server Options Service request messages use the OPTIONS method.

#### 6.8.1.1.2 Header Fields

The Retrieve Server Options Service request messages can include the following header fields:

- Accept:
  - application/vnd.sun.wadl+xml
  - application/json

#### 6.8.1.2 Response message

All responses are http single part messages. A successful response will return a Web Application Description Language (WADL) document encoded in a Media Type consistent with the Accept header of the request.

The WADL document shall contain one top-level "application" element.

The "application" element shall contain one "resources" element whose "base" attribute value is {SERVICE}, where {SERVICE} is the base URL for the service. This may be a combination of protocol (either http or https), host, port, and application.

Additionally, the WADL content shall include a "resource" element for the resource specified in the request (see 6.8.1.1) describing all methods (see 6.8.1.2.2.2 for description and examples) and child resources (see 6.8.1.2.2.1 for description and examples) for the specified resource and each of its children.

#### 6.8.1.2.1 Resources

The full WADL resource tree follows directly and unambiguously from the RESTful resource endpoints defined in 6.5, 6.6, 6.7 and 6.9.

For informative purposes, the full resource tree and the methods defined for each resource are described in Table 6.8-1.

**Table 6.8-1. Resources and Methods**

Resource	Methods supported (excluding RetrieveCapabilities)	Reference
{+SERVICE}	N/A	N/A
studies	SearchForStudies	6.8.1.2.2.3
	StoreInstances	6.8.1.2.2.2
{StudyInstanceUID}	RetrieveStudy	6.8.1.2.2.1
	StoreStudyInstances	6.8.1.2.2.3

Resource	Methods supported (excluding RetrieveCapabilities)	Reference
metadata	RetrieveStudyMetadata	6.8.1.2.2.1
series	SearchForStudySeries	6.8.1.2.2.3
{SeriesInstanceUID}	RetrieveSeries	6.8.1.2.2.1
metadata	RetrieveSeriesMetadata	6.8.1.2.2.1
instances	SearchForStudySeriesInstances	6.8.1.2.2.3
{SOPInstanceUID}	RetrieveInstance	6.8.1.2.2.1
metadata	RetrieveInstanceMetadata	6.8.1.2.2.1
frames	N/A	N/A
{framelist}	RetrieveFrames	6.8.1.2.2.1
instances	SearchForStudyInstances	6.8.1.2.2.3
series	SearchForSeries	6.8.1.2.2.3
{SeriesInstanceUID}	N/A	N/A
{instances}	SearchForInstances	6.8.1.2.2.3
instances	SearchForInstances	6.8.1.2.2.3
{BulkDataURL}	RetrieveBulkData	6.8.1.2.2.1
workitems	SearchForUPS	6.8.1.2.2.3
	CreateUPS	6.8.1.2.2.2
{UPSInstanceUID}	RetrieveUPS	6.8.1.2.2.1
	UpdateUPS	6.8.1.2.2.4
state	ChangeUPSSState	6.8.1.2.2.4
cancelrequest	RequestUPSCancel	6.8.1.2.2.4
subscribers	N/A	N/A
{AETitle}	CreateSubscription	6.8.1.2.2.5
	DeleteSubscription	6.8.1.2.2.5
1.2.840.10008.5.1.4.34.5	N/A	N/A
subscribers	N/A	N/A
{AETitle}	CreateSubscription	6.8.1.2.2.5
	DeleteSubscription	6.8.1.2.2.5
suspend	SuspendGlobalSubscription	6.8.1.2.2.5
1.2.840.10008.5.1.4.34.5.1	N/A	N/A
subscribers	N/A	N/A

Resource	Methods supported (excluding RetrieveCapabilities)	Reference
{AETitle}	CreateSubscription	6.8.1.2.2.5
	DeleteSubscription	6.8.1.2.2.5
suspend	SuspendGlobalSubscription	6.8.1.2.2.5

## 6.8.1.2.2 Methods

### 6.8.1.2.2.1 Retrieve Methods

The Retrieve methods define the capabilities of a WADO-RS resource (see 6.5) or a RetrieveUPS resource (see 6.9.4).

The Retrieve methods shall contain the following attributes:

- A "name" attribute with a value of "GET"
- An "id" attribute with a value of "RetrieveStudy", "RetrieveSeries", "RetrieveInstance", "RetrieveBulkData", "RetrieveFrames", "RetrieveStudyMetadata", "RetrieveSeriesMetadata", "RetrieveInstanceMetadata" or "RetrieveUPS"

The Retrieve methods shall contain a "request" element with "param" elements documenting the following:

- supported Accept header values
  - if the same Media Type is supported with multiple Transfer Syntaxes there should be one entry for each combination of Media Type and Transfer Syntax

The Retrieve methods shall contain one or more "response" elements documenting the following:

- supported Status Codes
- Media Types returned for each Status Code (if applicable)
  - if the same Media Type is supported with multiple Transfer Syntaxes there should be one entry for each combination of Media Type and Transfer Syntax

#### Note

More than one Status Code can be described by a single "response" element.

Example:

```
<method name="GET" id="RetrieveStudies">
  <request>
    <param name="Accept" style="header" default="multipart/related; type=application/dicom">
      <option value="multipart/related; type=application/dicom" />
      <option value="multipart/related; type=application/dicom";
        transfer-syntax=1.2.840.10008.1.2 />
      <option value="multipart/related; type=application/dicom";
        transfer-syntax=1.2.840.10008.1.2.1 />
      <option value="multipart/related; type=application/octet-stream" />
      <option value="multipart/related; type=image/dicom+jpx" />
      <option value="multipart/related; type=image/dicom+jpx;
        transfer-syntax=1.2.840.10008.1.2.4.92" />
      <option value="multipart/related; type= video/mpeg;
        transfer-syntax=1.2.840.10008.1.2.4.100" />
    </param>
  </request>
  <response status="200,206">
    <representation mediaType="multipart/related; type=application/dicom";
```

```

        transfer-syntax=1.2.840.10008.1.2 />
<representation mediaType="multipart/related; type=application/dicom";
    transfer-syntax=1.2.840.10008.1.2.1 />
<representation mediaType="multipart/related; type=application/octet-stream" />
<representation mediaType="multipart/related; type= image/dicom+jpx" />
<representation mediaType="multipart/related; type= image/dicom+jpx;
    transfer-syntax=1.2.840.10008.1.2.4.92" />
<representation mediaType="multipart/related; type= video/mpeg;
    transfer-syntax=1.2.840.10008.1.2.4.100" />
</response>
<response status="400 404 406 410 503" />
</method>

```

#### 6.8.1.2.2.2 Store Methods

The Store methods define the capabilities of a STOW-RS resource (see 6.6) or a CreateUPS resource (see 6.9.1).

The Store methods shall contain the following attributes:

- A "name" attribute with a value of "POST"
- An "id" attribute with a value of "StoreInstances", "StoreStudyInstances" or "CreateUPS"

The Store methods shall contain a "request" element with "param" elements documenting the following:

- supported Accept header values
- supported Representations
  - if the same Media Type is supported with multiple Transfer Syntaxes there should be one entry for each combination of Media Type and Transfer Syntax

The Store methods shall contain one or more "response" elements documenting the following:

- supported Status Codes
- Media Types returned for each Status Code (if applicable)
- Headers returned for each Status Code

#### Note

More than one Status Code can be described by a single "response" element.

Example:

```

<method name="GET" id="StoreInstances">
<request>
<param name="Accept" style="header" default="application/dicom+xml">
    <option value="application/dicom+xml" />
</param>
<representation mediaType="multipart/related; type=application/dicom" />
<representation mediaType="multipart/related; type=application/dicom;
    transfer-syntax=1.2.840.10008.1.2" />
<representation mediaType="multipart/related; type=application/dicom;
    transfer-syntax=1.2.840.10008.1.2.1" />
<representation mediaType="multipart/related; type=application/dicom+xml" />
<representation mediaType="multipart/related; type=application/dicom+xml;
    transfer-syntax=1.2.840.10008.1.2" />
<representation mediaType="multipart/related; type=application/dicom+xml;
    transfer-syntax=1.2.840.10008.1.2.1" />

```

```

<representation mediaType="multipart/related; type=application/dicom+xml;
  transfer-syntax=1.2.840.10008.1.2.4.92" />
<representation mediaType="multipart/related; type=application/dicom+xml;
  transfer-syntax=1.2.840.10008.1.2.4.100" />
</request>
<response status="200" />
<response status="202,409">
  <representation mediaType="application/dicom+xml" />
</response>
<response status="400,401,403,503" />
</method>

```

### 6.8.1.2.2.3 Search Methods

The Search methods define the capabilities of a QIDO-RS resource (see 6.7) or a SearchForUPS resource (see 6.9.3).

The Search methods shall contain the following attributes:

- A "name" attribute with a value of "GET"
- An "id" attribute with a value of "SearchForStudies", "SearchForStudySeries", "SearchForSeries", "SearchForStudySeriesInstances", "SearchForStudyInstances", "SearchForSeriesInstances", "SearchForInstances" or "SearchForUPS"

The Search methods shall contain a "request" element with "param" elements documenting the following:

- supported Accept header values
- support for the Cache-control header
- support of "limit", "offset" and "fuzzymatching" query parameters
- supported search parameters (both tag and keyword variants shall be listed)
- supported options for the "includefield" parameter (both tag and keyword variants shall be listed)

The Search methods shall contain one or more "response" elements documenting the following:

- supported Status Codes
- returned "header" parameters, including use of "warning headers"
- Media Types returned for each Status Code (if applicable)

#### Note

More than one Status Code can be described by a single "response" element.

Example:

```

<method name="GET" id="SearchForStudies">
  <request>
    <param name="Accept" style="header" default="multipart/related; type=application/dicom+xml">
      <option value="multipart/related; type=application/dicom+xml" />
      <option value="application/json" />
    </param>
    <param name="Cache-control" style="header">
      <option value="no-cache" />
    </param>
    <param name="limit" style="query" />
    <param name="offset" style="query" />
    <param name="fuzzymatching" style="query" />
    <param name="StudyDate" style="query" />
  </request>

```

```

<param name="00080020" style="query" />
<param name="StudyTime" style="query" />
<param name="00080030" style="query" />
...
<param name="includefield" style="query" repeating="true" />
<option value="all" />
<option value="00081049" />
<option value="PhysiciansOfRecordIdentificationSequence" />
<option value="00081060" />
<option value="NameOfPhysiciansReadingStudy" />
...
</param>
</request>
<response status="200">
<representation mediaType="multipart/related; type=application/dicom+xml" />
<representation mediaType="application/json" />
</response>
<response status="400 401 403 413 503" />
</method>

```

#### 6.8.1.2.2.4 Update Methods

The Update methods define the capabilities of an UpdateUPS, a ChangeUPSSState or a RequestUPSCancellation resource (see 6.9.2).

The Update methods shall contain the following attributes:

- A "name" attribute with a value of "POST" for UpdateUPS and RequestUPSCancel
- A "name" attribute with a value of "PUT" for ChangeUPSSState
- An "id" attribute with a value of "UpdateUPS", "ChangeUPSSState" or "RequestUPSCancellation"

The Update methods shall contain a "request" element with "param" elements documenting the following:

- supported Representations

The Update methods shall contain one or more "response" elements documenting the following:

- supported Status Codes
- Headers returned for each Status Code

Note

More than one Status Code can be described by a single "response" element.

Example:

```

<method name="POST" id="UpdateUPS">
<request>
<representation mediaType="application/dicom+xml" />
<representation mediaType="application/json" />
</request>
<response status="200">
<param name="Warning" style="header" fixed="299 {+SERVICE}: The UPS was created with modifications." />
<param name="Warning" style="header" fixed="299 {+SERVICE}: Requested optional Attributes are not supported." />
</response>
<response status="409">
<param name="Warning" style="header" fixed="299 {+SERVICE}: The Transaction UID is missing." />
<param name="Warning" style="header" fixed="299 {+SERVICE}: The Transaction UID is incorrect." />

```

```

    <param name="Warning" style="header" fixed="299 {+SERVICE}: The submitted request is inconsistent with the current state of the UPS
  </response>
  <response status="400 401 403 404 503" />
</method>

```

#### 6.8.1.2.2.5 Subscribe Methods

The Subscribe methods define the capabilities of a CreateSubscription, a SuspendGlobalSubscription or a DeleteSubscription resource (see 6.9.7, 6.9.8 and 6.9.9).

The Subscribe methods shall contain the following attributes:

- A "name" attribute with a value of "POST" for CreateSubscription and SuspendGlobalSubscription
- A "name" attribute with a value of "DELETE" for DeleteSubscription
- An "id" attribute with a value of "CreateSubscription", "SuspendGlobalSubscription" or "DeleteSubscription"

The Subscribe methods shall contain a "request" element with "param" elements documenting the following:

- supported Representations

The Subscribe methods shall contain one or more "response" elements documenting the following:

- supported Status Codes
- Headers returned for each Status Code

Note

More than one Status Code can be described by a single "response" element.

Example:

```

<method name="POST" id="CreateSubscription">
  <request>
    <param name="deletionlock" style="query" default="false">
      <option value="true" />
      <option value="false" />
    </param>
  </request>
  <response status="201">
    <param name="Warning" style="header" fixed="299 {+SERVICE}: Deletion Lock not granted." />
  </response>
  <response status="403">
    <param name="Warning" style="header" fixed="299 {+SERVICE}: The Origin-Server does not support Global Subscription Filtering." />
  </response>
  <response status="400 401 404 409 503" />
</method>

```

#### 6.8.1.3 Status Codes

Table 6.8-2 lists the HTTP/1.1 status codes that shall be used to report any of the associated error and warning situations. Other error codes may be present for other error and warning situations.

**Table 6.8-2. Server Options HTTP/1.1 Status Codes**

Code	Name	Description
Success		

Code	Name	Description
200	OK	The query completed and any matching results are returned in the message body.
Failure		
400	Bad Request	The Server Options Provider was unable to perform the query because the Service Provider cannot understand the query component.
401	Unauthorized	The Server Options Provider refused to perform the query because the client is not authenticated.
403	Forbidden	The Server Options Provider understood the request, but is refusing to perform the query (e.g., an authenticated user with insufficient privileges).
503	Busy	Service is unavailable.

## 6.9 UPS-RS Worklist Service

This DICOM Web Service defines a RESTful interface to the UPS SOP Classes (see PS3.3 & PS3.4). It consists of the following action types:

### 1. CreateUPS

This action requests the creation of a UPS Instance on the Origin-Server. It corresponds to the UPS DIMSE N-CREATE operation.

### 2. UpdateUPS

This action sets the attributes of a UPS Instance managed by the Origin-Server. It corresponds to the UPS DIMSE N-SET operation.

### 3. SearchForUPS

This action searches for UPS Instances known to the Origin-Server. It corresponds to the UPS DIMSE C-FIND operation.

### 4. RetrieveUPS

This action retrieves a UPS Instances. It corresponds to the UPS DIMSE N-GET operation.

### 5. ChangeUPSState

This action sets the state of a UPS Instance managed by the Origin-Server. It corresponds to the UPS DIMSE N-ACTION operation "Change UPS State".

### 6. RequestUPSCancellation

This action requests the cancellation of a UPS Instance managed by the Origin-Server. It corresponds to the UPS DIMSE N-ACTION operation "Request UPS Cancel".

### 7. CreateSubscription

This action subscribes to a UPS Instance or the Global Worklist managed by the Origin-Server. It corresponds to the UPS DIMSE N-ACTION operation "Subscribe to Receive UPS Event Reports".

### 8. SuspendGlobalSubscription

This action suspends an existing subscription to the Global Worklist managed by the Origin-Server. It corresponds to the UPS DIMSE N-ACTION operation "Suspend Global Subscription".

### 9. DeleteSubscription

This action cancels an existing subscription to a UPS Instance or the Global Worklist managed by the Origin-Server. It corresponds to the UPS DIMSE N-ACTION operation "Unsubscribe from Receiving UPS Event Reports".

### 10. OpenEventChannel

This action initiates a WebSocket connection to allow the User-Agent to start receiving Event Report messages.

#### 11. SendEventReport

This action sends an Event Report using an open WebSocket connection. It corresponds to the UPS DIMSE N-EVENT-REPORT operation.

An Origin-Server shall support all of the above action types.

The requirements for a UPS-RS Origin-Server that is also a Unified Worklist and Procedure Step SCP are described in Section CC.1 in PS3.4

**Table 6.9-1. UPS Interface Mapping**

Action Type	Section	Method & Resource
CreateUPS	6.9.1	POST {+SERVICE}/workitems{?AffectedSOPInstanceUID}
UpdateUPS	6.9.2	POST {+SERVICE}/workitems/{UPSInstanceUID}{?transaction}
SearchForUPS	6.9.3	GET {+SERVICE}/workitems{?query*}
RetrieveUPS	6.9.4	GET {+SERVICE}/workitems/{UPSInstanceUID}
ChangeUPSSState	6.9.5	PUT {+SERVICE}/workitems/{UPSInstanceUID}/state
RequestUPSCancellation	6.9.6	POST {+SERVICE}/workitems/{UPSInstanceUID}/cancelrequest
CreateSubscription	6.9.7	POST {+SERVICE}/workitems/{UPSInstanceUID}/subscribers/{AETitle}{?deletionlock}  POST {+SERVICE}/workitems/1.2.840.10008.5.1.4.34.5/ subscribers/{AETitle}{?deletionlock}  POST {+SERVICE}/workitems/1.2.840.10008.5.1.4.34.5.1/ subscribers/{AETitle}{?deletionlock,query*}
SuspendGlobalSubscription	6.9.8	POST {+SERVICE}/workitems/1.2.840.10008.5.1.4.34.5/ subscribers/{AETitle}/suspend  POST {+SERVICE}/workitems/1.2.840.10008.5.1.4.34.5.1/ subscribers/{AETitle}/suspend
DeleteSubscription	6.9.9	DELETE {+SERVICE}/workitems/{UPSInstanceUID}/ subscribers/{AETitle}
OpenEventChannel	6.9.10	GET {+WSSERVICE}/subscribers/{AETitle}
SendEventReport	6.9.11	N/A

The Origin-Server shall comply with all requirements placed on the SCP for the corresponding services in Annex CC in PS3.4.

### 6.9.1 CreateUPS

This resource allows a User-Agent to instruct an Origin-Server to create a UPS instance.

#### 6.9.1.1 Request

The request message shall be formed as follows:

- Resource
  - {+SERVICE}/workitems{?AffectedSOPInstanceUID}

where

- {+SERVICE} is the base URL for the service. This may be a combination of protocol (either HTTP or HTTPS), authority and path.
- {AffectedSOPInstanceUID} specifies the SOP Instance UID of the UPS Instance to be created
- Method
  - POST
- Headers
  - Content-Type - The representation scheme being posted to the RESTful service. The types allowed for this request header are as follows:
    - application/dicom+xml  
Specifies that the post is DICOM PS3.19 XML metadata. See 6.9.1.1.1.
    - application/json  
Specifies that the post is DICOM PS3.18 JSON metadata. See 6.9.1.1.1.
  - The request body shall convey a single Unified Procedure Step Instance. The instance shall comply with all requirements in the Req. Type N-CREATE column of Table CC.2.5-3 in PS3.4.

#### 6.9.1.1.1 Request Message

The Request Message has a single part body.

- Content-Type:
  - application/dicom+xml
  - application/json
- The request body contains all attributes to be stored in either DICOM PS3.19 XML or DICOM PS3.18 JSON. Any binary data contained in the message shall be inline.

#### 6.9.1.2 Behavior

The Origin-Server shall create and maintain UPS instances as instructed by CreateUPS requests and as specified by the SCP behavior in Section CC.2.5.3 in PS3.4.

The Origin-Server shall return the HTTP/1.1 Status Line applicable to the associated request.

#### 6.9.1.3 Response

The Origin-Server shall return an HTTP/1.1 response message.

##### 6.9.1.3.1 Response Status Line

If the Create request is successful, the Origin-Server shall return an HTTP/1.1 "201 - Created" response code.

If the request fails, the Origin-Server shall return an appropriate failure status line with a response code from 6.9.1-1.

**Table 6.9.1-1. Status Codes**

HTTP/1.1 Code	Reason Phrase	Description
201	Created	The UPS instance was created and the new resource can be retrieved at the Content-Location specified in the response

HTTP/1.1 Code	Reason Phrase	Description
400	Bad Request	The UPS-RS Origin-Server was unable to understand the request
401	Unauthorized	The UPS-RS Origin-Server refused to accept the request because the client is not authenticated.
403	Forbidden	The UPS-RS Origin-Server understood the request, but is refusing to perform the query (e.g., an authenticated user with insufficient privileges).
409	Conflict	The UID of the posted UPS Instance corresponds to an existing UPS Instance.
503	Busy	Service is unavailable.

### 6.9.1.3.2 Response Headers

If the request is successful, the HTTP/1.1 response message shall include the following HTTP/1.1 header:

- Content-Location: {+WorkitemURL}

Where {+WorkitemURL} is the URL from which the created UPS Instance can be retrieved (see 6.9.4)

If the UPS instance was created with modifications, the response message shall include the following HTTP/1.1 header:

- Warning: 299 {+SERVICE}: The UPS was created with modifications.

### 6.9.1.3.3 Response Message Body

The response message body shall be empty.

## 6.9.2 UpdateUPS

This resource supports the modification of attribute values of an existing UPS Instance.

### 6.9.2.1 Request

The request message shall be formed as follows:

- Resource
  - {+SERVICE}/workitems/{UPSInstanceUID}{?transaction}

where

  - {+SERVICE} is the base URL for the service. This may be a combination of protocol (either HTTP or HTTPS), authority and path.
  - {UPSInstanceUID} is the UID of the Unified Procedure Step Instance
  - {transaction} specifies the Transaction UID / Locking UID for the specified Unified Procedure Step Instance

If the UPS instance is currently in the SCHEDULED state, {transaction} shall not be specified.

If the UPS instance is currently in the IN PROGRESS state, {transaction} shall be specified.
- Method
  - POST
- Headers
  - Content-Type - The representation scheme being posted to the RESTful service. The types allowed for this request header are as follows:
    - application/dicom+xml

Specifies that the post is DICOM PS3.19 XML metadata. See Section 6.9.2.1.1.

- application/json

Specifies that the post is DICOM PS3.18 JSON metadata. See Section 6.9.2.1.1.

- The request body describes changes to a single Unified Procedure Step Instance. It shall include all Attributes for which Attribute Values are to be set. The changes shall comply with all requirements described in Section CC.2.6.2 in PS3.4.
- Because the request will be treated as atomic (indivisible) and idempotent (repeat executions have no additional effect), all changes contained in the request shall leave the UPS instance in an internally consistent state.

### 6.9.2.1.1 Request Message

The Request Message has a single part body.

- Content-Type:
  - application/dicom+xml
  - application/json
- The request body contains all the attributes to be updated in either DICOM PS3.19 XML or DICOM PS3.18 JSON. Any binary data contained in the message shall be inline.

### 6.9.2.2 Behavior

The Origin-Server shall support the Attribute changes to the UPS instance specified by the User-Agent in the UpdateUPS request and as specified by the SCP behavior in Section CC.2.6.3 in PS3.4.

The Origin-Server shall return the HTTP/1.1 Status applicable to the associated request.

### 6.9.2.3 Response

The Origin-Server shall return an HTTP/1.1 response message.

#### 6.9.2.3.1 Response Status Line

If the Set request is successful, the Origin-Server shall return an HTTP/1.1 "200 - OK" response code.

If the request fails, the Origin-Server shall return an appropriate failure status line with a response code from Table 6.9.2-1, "Status Codes".

**Table 6.9.2-1. Status Codes**

HTTP/1.1 Code	Reason Phrase	Description
200	OK	The UPS instance was updated
400	Bad Request	The UPS-RS Origin-Server was unable to understand the request
401	Unauthorized	The UPS-RS Origin-Server refused to accept the request because the client is not authenticated.
403	Forbidden	The UPS-RS Origin-Server understood the request, but is refusing to perform the query (e.g., an authenticated user with insufficient privileges).
404	Not found	The specified UPS Instance does not exist or is not managed by this Origin-Server.

HTTP/1.1 Code	Reason Phrase	Description
409	Conflict	The request cannot be performed for one of the following reasons: <ul style="list-style-type: none"> <li>the submitted request is inconsistent with the current state of the UPS Instance</li> <li>the Transaction UID is missing</li> <li>the Transaction UID is incorrect</li> </ul>
503	Busy	Service is unavailable.

### 6.9.2.3.2 Response Headers

If the UPS instance was updated but with modifications made by the Origin-Server, the response message shall include the following HTTP/1.1 header:

- Warning: 299 {+SERVICE}: The UPS was created with modifications.

If optional attributes were rejected, the response message shall include the following HTTP/1.1 Warning header field:

- Warning: 299 {+SERVICE}: Requested optional Attributes are not supported.

If the request was rejected with an HTTP/1.1 409 status code, the response message shall include one of following messages encoded in an HTTP/1.1 Warning header field describing the nature of the conflict:

- Warning: 299 {+SERVICE}:The Transaction UID is missing.
- Warning: 299 {+SERVICE}:The Transaction UID is incorrect.
- Warning: 299 {+SERVICE}:The submitted request is inconsistent with the current state of the UPS Instance.

### 6.9.2.3.3 Response Message Body

The response message body shall be empty.

## 6.9.3 SearchForUPS

This resource returns a list of UPS Instances that match specified search query parameters along with requested attributes for each Instance.

### 6.9.3.1 Request

The request message shall be formed as follows:

- Resource
  - {+SERVICE}/workitems/{?query\*}

where

  - {+SERVICE} is the base URL for the service. This may be a combination of protocol (either HTTP or HTTPS), authority and path.
- Method
  - GET
- Headers
  - Accept - The representation scheme in which the RESTful service is requested to return the results. The types allowed for this request header are as follows:

- multipart/related; type=application/dicom+xml; boundary={messageBoundary}

Specifies that the results should be DICOM PS3.19 XML metadata.

- application/json

Specifies that the results should be DICOM PS3.18 JSON metadata.

- Cache-control: no-cache (recommended)

If included, specifies that search results returned should be current and not cached.

- {query}

- {attributeID}={value}

0-n / {attributeID}={value} pairs allowed

- includefield={attributeID} | all

0-n includefield / {attributeID} pairs allowed, where "all" indicates that all attributes with values should be included for each response.

Each {attributeID} shall refer to an attribute of the Unified Procedure Step IOD (see Section B.26.2 in PS3.3).

See Section 6.7.1.1 for {attributeID} and {value} encoding rules

- fuzzymatching=true | false

- limit={maximumResults}

- offset={skippedResults}

### 6.9.3.2 Behavior

The Origin-Server shall perform a search according the requirements for the QIDO-RS services (see Section 6.7.1.2).

#### 6.9.3.2.1 Matching

An Origin-Server shall support matching against all Unified Procedure Step Instance Attributes in Table CC.2.5-3 in PS3.4 with a Match Key Type value of U, R or \*.

See Section 6.7.1.2.1 for matching behavior.

### 6.9.3.3 Response

The Origin-Server shall return an HTTP/1.1 response message.

#### 6.9.3.3.1 Response Status Line

If the SearchForUPS request is successful, the Origin-Server shall return an HTTP/1.1 "200 - OK" response code.

If the request fails, the Origin-Server shall return an appropriate failure status line with a response code from Table 6.9.3-1, "Status Codes".

**Table 6.9.3-1. Status Codes**

HTTP/1.1 Code	Reason Phrase	Description
200	OK	The query completed and any matching results are returned in the message body.
206	Partial Content	Only some of the query results were returned and the rest can be requested through the appropriate UPS-RS request.

HTTP/1.1 Code	Reason Phrase	Description
400	Bad Request	The UPS-RS Origin-Server was unable to perform the query because the Service Provider cannot understand the query component.
401	Unauthorized	The UPS-RS Origin-Server refused to perform the query because the client is not authenticated.
403	Forbidden	The UPS-RS Origin-Server understood the request, but is refusing to perform the query (e.g., an authenticated user with insufficient privileges).
413	Request entity too large	The query was too broad and a narrower query or paging should be requested.
503	Busy	Service is unavailable.

### 6.9.3.3.2 Query Result Attribute

For each matching UPS Instance, the Origin-Server shall return:

- -All Unified Procedure Step Instance Attributes in Table CC.2.5-3 in PS3.4 with a Return Key value of 1 and 2.
- -All Unified Procedure Step Instance Attributes in Table CC.2.5-3 in PS3.4 with a Return Key value of 1C for which the conditional requirements are met.
- -All other Unified Procedure Step Instance Attributes passed as {attributeID} query keys that are supported by the Origin-Server as matching or return attributes
- -All other Unified Procedure Step Instance Attributes passed as "includefield" query values that are supported by the Origin-Server as return attributes.

### 6.9.3.3.3 Response Message

The response message body contains the results.

The format of the response message body shall contain one of the Media Types specified by the request Accept header field. An Origin-Server shall support all Media-Types allowed in the request.

#### 6.9.3.3.3.1 XML Response Message

- Content-Type:
  - multipart/related; type=application/dicom+xml
- The response is a multipart message body where each part is a DICOM PS3.19 XML DicomNativeModel element containing the attributes for one matching UPS Instance (see Section A.1 in PS3.19).
- If there are no matching results, the message body shall be empty.
- -Each part in the multipart body includes the following HTTP/1.1 headers:
  - Content-Type: application/dicom+xml

#### 6.9.3.3.3.2 JSON Response Message

- Content-Type:
  - application/json
- The response is a DICOM JSON message containing a DICOM JSON property for each matching UPS Instance containing sub-properties describing the matching attributes for each UPS Instance (see Section F.2).
- If there are no matching results, the JSON message shall be empty.

## 6.9.4 RetrieveUPS

This resource supports the retrieval of a UPS Instance.

### 6.9.4.1 Request

The request message shall be formed as follows:

- Resource
  - {+SERVICE}/workitems/{UPSInstanceUID}
  - where
    - {+SERVICE} is the base URL for the service. This may be a combination of protocol (either HTTP or HTTPS), authority and path.
    - {UPSInstanceUID} is the UID of the Unified Procedure Step Instance
- Method
  - GET
- Headers
  - Accept - The representation scheme in which the RESTful service is requested to return the result. The types allowed for this request header are as follows:
    - application/dicom+xml
      - Specifies that the result should be DICOM PS3.19 XML metadata.
    - application/json
      - Specifies that the result should be DICOM PS3.18 JSON metadata.
  - Cache-control: no-cache (recommended)
    - If included, specifies that results returned should be current and not cached.

### 6.9.4.2 Behavior

The Origin-Server shall return, via the HTTP/1.1 response, the indicated Unified Procedure Step Instance to the User-Agent.

Note

The requirement for the Origin-Server to respond to GET requests for UPS Instances that have moved to the COMPLETED or CANCELED state is limited. See Section CC.2.1.3 in PS3.4.

The User-Agent shall not return the Transaction UID (0008,1195) Attribute. This is necessary to preserve this Attribute's role as an access lock.

The User-Agent shall return the HTTP/1.1 Response Status Code applicable to the associated request. A Failure Code shall indicate that the Origin-Server has not returned the SOP Instance.

### 6.9.4.3 Response

The Origin-Server shall return an HTTP/1.1 response message.

#### 6.9.4.3.1 Response Status Line

If the Retrieve request is successful, the Origin-Server shall return an HTTP/1.1 "200 - OK" response code.

If the request fails, the Origin-Server shall return an appropriate failure status line with a response code from Table 6.9.4-1, "Status Codes".

**Table 6.9.4-1. Status Codes**

HTTP/1.1 Code	Reason Phrase	Description
200	OK	The requested instance is returned.
400	Bad Request	The UPS-RS Origin-Server was unable to perform the query because the Service Provider cannot understand the query component.
401	Unauthorized	The UPS-RS Origin-Server refused to perform the query because the client is not authenticated.
403	Forbidden	The UPS-RS Origin-Server understood the request, but is refusing to perform the query (e.g., an authenticated user with insufficient privileges).
404	Not found	The specified UPS Instance does not exist or is not managed by this Origin-Server.
503	Busy	Service is unavailable.

### 6.9.4.3.2 Response Message

The response message body contains the results.

The format of the response message body shall contain one of the Media Types specified by the request Accept header field. An Origin-Server shall support all Media-Types allowed in the request.

#### 6.9.4.3.2.1 XML Response Message

- Content-Type:
  - application/dicom+xml
- - The response contains a DICOM PS3.19 XML DicomNativeModel element containing the attributes for the requested UPS Instance (see Section A.1 in PS3.19).

#### 6.9.4.3.2.2 JSON Response Message

- Content-Type:
  - application/json
- The response is a DICOM JSON array containing a DICOM JSON representation of the requested UPS Instance (see Section F.2).

## 6.9.5 ChangeUPSState

This resource supports the modification of the state of an existing UPS Instance.

### 6.9.5.1 Request

The request message shall be formed as follows:

- Resource
    - {+SERVICE}/workitems/{UPSInstanceUID}/state
- where:
- {+SERVICE} is the base URL for the service. This may be a combination of protocol (either HTTP or HTTPS), authority and path.
  - {UPSInstanceUID} is the UID of the Unified Procedure Step Instance

- Method
  - PUT
- Headers
  - Content-Type - The representation scheme being posted to the RESTful service. The types allowed for this request header are as follows:
    - application/dicom+xml  
Specifies that the post is DICOM PS3.19 XML metadata. See Section 6.9.5.1.1.
    - application/json  
Specifies that the post is DICOM PS3.18 JSON metadata. See Section 6.9.5.1.1.
  - The request body describes a state change to a single Unified Procedure Step Instance. It shall include all Attributes required for an SCU in Table CC.2.1-1 in PS3.4.

### 6.9.5.1.1 Request Message

The Request Message has a single part body.

- Content-Type:
  - application/dicom+xml
  - application/json
- The request body contains attributes in either DICOM PS3.19 XML or DICOM PS3.18 JSON format.

### 6.9.5.2 Behavior

The Origin-Server shall support the state changes to the UPS instance specified in the request as described by the SCP behavior in Section CC.2.1.3 in PS3.4.

After completing the ChangeUPSSState request, the Origin-Server shall return the HTTP/1.1 Response Line applicable to the associated request.

### 6.9.5.3 Response

The Origin-Server shall return an HTTP/1.1 response message.

#### 6.9.5.3.1 Response Status Line

If the State Change was successful, the Service shall return an HTTP/1.1 "200 - OK" response code.

If the State Change fails, the Service shall return an appropriate failure status line with a response code from Table 6.9.5-1, "Status Codes".

**Table 6.9.5-1. Status Codes**

HTTP/1.1 Code	Reason Phrase	Description
200	OK	The UPS instance was updated
400	Bad Request	The UPS-RS Origin-Server was unable to understand the request
401	Unauthorized	The UPS-RS Origin-Server refused to accept the request because the client is not authenticated.
403	Forbidden	The UPS-RS Origin-Server understood the request, but is refusing to perform the query (e.g., an authenticated user with insufficient privileges).

HTTP/1.1 Code	Reason Phrase	Description
404	Not found	The specified UPS Instance does not exist or is not managed by this Origin-Server.
409	Conflict	The request cannot be performed for one of the following reasons: <ul style="list-style-type: none"> <li>• the submitted request is inconsistent with the current state of the UPS Instance</li> <li>• the Transaction UID is missing</li> <li>• the Transaction UID is incorrect</li> </ul>
503	Busy	Service is unavailable.

### 6.9.5.3.2 Response Headers

If the User-Agent specifies a Procedure Step State (0074,1000) attribute with a value of "CANCELED" and the UPS Instance is already in that state, the response message shall include the following HTTP/1.1 Warning header field:

- Warning: 299 {+SERVICE}: The UPS is already in the requested state of CANCELED.

If the User-Agent specifies a Procedure Step State (0074,1000) attribute with a value of "COMPLETED" and the UPS Instance is already in that state, the response message shall include the following HTTP/1.1 Warning header field:

- Warning: 299 {+SERVICE}: The UPS is already in the requested state of COMPLETED.

If the request was rejected with an HTTP/1.1 409 status code, the response message shall include one of following messages in the HTTP/1.1 Warning header field describing the nature of the conflict:

- Warning: 299 {+SERVICE}:the Transaction UID is missing.
- Warning: 299 {+SERVICE}:the Transaction UID is incorrect.
- Warning: 299 {+SERVICE}:the submitted request is inconsistent with the current state of the UPS Instance.

### 6.9.5.3.3 Response Message Body

The response message body shall be empty.

## 6.9.6 RequestUPSCancellation

This resource records a request that the specified UPS Instance be canceled.

### 6.9.6.1 Request

- Resource
  - `-{+SERVICE}/workitems/{UPSInstanceUID}/cancelrequest`

where:

    - {+SERVICE} is the base URL for the service. This may be a combination of protocol (either HTTP or HTTPS), authority and path.
    - {UPSInstanceUID} is the UID of the Unified Procedure Step Instance
- Method
  - POST
- Headers

- Content-Type - The representation scheme being posted to the RESTful service. The types allowed for this request header are as follows:

- application/dicom+xml

Specifies that the post is DICOM PS3.19 XML metadata. See Section 6.9.5.1.1.

- application/json

Specifies that the post is DICOM PS3.18 JSON metadata. See Section 6.9.5.1.1.

- The request body describes a request to cancel a single Unified Procedure Step Instance. The request body shall comply with all attribute requirements described in Table CC.2.2-1 in PS3.4.

### 6.9.6.1.1 Request Message

The Request Message has a single part body.

- Content-Type:

- application/dicom+xml

- application/json

- The request body contains attributes in either DICOM PS3.19 XML or DICOM PS3.18 JSON format.

### 6.9.6.2 Behavior

RequestUPSCancellation is used to request to the Origin-Server that the state of a UPS Instance be changed to CANCELED as shown in Figure CC.1.1-1 in PS3.4. The Origin-Server shall process the request as described by the SCP behavior in Section CC.2.2.3 in PS3.4.

The request may include a Reason For Cancellation and/or a proposed Procedure Step Discontinuation Reason Code Sequence.

The request may also include a Contact Display Name and/or a Contact URI for the person with whom the cancel request may be discussed.

#### Note

An HTTP/1.1 Status Code indicating success means that the Request was accepted, not that the UPS has been canceled. The system performing the UPS is not obliged to honor the request to cancel and in some scenarios, may not even receive notification of the request. See Section CC.2.4 in PS3.4.

To cancel an IN PROGRESS UPS that the User-Agent is itself performing, the User-Agent shall instead use the ChangeUPSSState action as described in Section 6.9.5, "ChangeUPSSState".

### 6.9.6.3 Response

The Origin-Server shall return an HTTP/1.1 response message.

#### 6.9.6.2.1 Response Status Line

If the cancel request was accepted, the Service shall return an HTTP/1.1 "202 - Accepted" response code.

If the cancel request was rejected, the Service shall return an appropriate failure status line with a response code from Table 6.9.6-1, "Status Codes".

**Table 6.9.6-1. Status Codes**

HTTP/1.1 Code	Reason Phrase	Description
202	Accepted	The cancel request was accepted

HTTP/1.1 Code	Reason Phrase	Description
400	Bad Request	The UPS-RS Origin-Server was unable to understand the request
401	Unauthorized	The UPS-RS Origin-Server refused to accept the request because the client is not authenticated.
403	Forbidden	The UPS-RS Origin-Server understood the request, but is refusing to perform the query (e.g., an authenticated user with insufficient privileges).
404	Not found	The specified UPS Instance does not exist or is not managed by this Origin-Server.
409	Conflict	The cancellation request is inconsistent with the current state of the UPS Instance
503	Busy	Service is unavailable.

### 6.9.2.5.2 Response Headers

If the UPS Instance is already in a canceled state, the response message shall include the following HTTP/1.1 Warning header field:

- Warning: 299 {+SERVICE}: The UPS is already in the requested state of CANCELED.

### 6.9.5.2.3 Response Message Body

The response message body shall be empty.

## 6.9.7 CreateSubscription

This resource records subscribers to whom future events associated with the specified UPS Instances will be reported.

### 6.9.7.1 Request

The request message shall be formed as follows:

- Resource
  - {+SERVICE}/workitems/{UPSInstanceUID}/subscribers/{AETitle}{?deletionlock}
  - {+SERVICE}/workitems/1.2.840.10008.5.1.4.34.5/subscribers/{AETitle}{?deletionlock}
  - {+SERVICE}/workitems/1.2.840.10008.5.1.4.34.5.1/subscribers/{AETitle}{?deletionlock,query\*}

where

- {+SERVICE} is the base URL for the service. This may be a combination of protocol (either HTTP or HTTPS), authority and path.
- {UPSInstanceUID} is the UID of the Unified Procedure Step Instance or a well-known UID
- {AETitle} is an Application Entity Title that conforms to the "AE" Value Representation (see Table 6.2-1 in PS3.5) and identifies the Application Entity to be subscribed
- {deletionlock}, if present, shall have a value of either "true" or "false", indicating whether or not the User-Agent is requesting a Deletion Lock
- {query} specifies the query key/value pairs describing the filter parameters
- Method
  - POST
- Headers

- Content-Length: 0
- {query}
  - deletionlock=true | false
- {attributeID}={value}
  - 0-n / {attributeID}={value} pairs allowed

Each {attributeID} shall refer to an attribute of the Unified Procedure Step IOD (see Section B.26.2 in PS3.3).

See Section 6.7.1.1 for {attributeID} and {value} encoding rules.

- The request body shall be empty.

## 6.9.7.2 Behavior

The Origin-Server shall support the management of UPS instance subscriptions as specified by the SCP behavior in Section CC.2.3.3 in PS3.4.

Upon receipt of the CreateSubscription, SuspendGlobalSubscription or DeleteSubscription request, the Origin-Server shall attempt to update the Global Subscription State, Filtered Global Subscription and/or UPS Subscription State of the specified Application Entity with respect to the specified SOP Instance UID as described in Table CC.2.3-2 in PS3.4 and then return the appropriate HTTP/1.1 response.

## 6.9.7.3 Response

### 6.9.7.3.1 Response Status Line

The Service shall return an HTTP/1.1 status line, including a status code and associated reason phrase.

If the CreateSubscription request was successful, the Service shall return an "HTTP/1.1 201 - Created" response code. The response shall contain a "Content-Location" header of the following format:

- Content-Location: {WSSERVICE}

where:

- - {WSSERVICE} is the base URL for the WebSocket service. This shall include the WebSocket protocol (either WS or WSS) and may include a combination of authority and path.

If the subscription fails, the Service shall return an appropriate failure status line with a response code from Table 6.9.7-2, "Status Codes".

**Table 6.9.7-2. Status Codes**

HTTP/1.1 Code	Reason Phrase	Description
201	Created	The subscription was created.
400	Bad Request	The UPS-RS Origin-Server was unable to understand the request
401	Unauthorized	The UPS-RS Origin-Server refused to accept the request because the client is not authenticated.
403	Forbidden	The UPS-RS Origin-Server understood the request, but is refusing to perform the query (e.g., the Origin-Server does not support global subscription filtering or an authenticated user has insufficient privileges).
404	Not found	The specified UPS Instance or well-known UID does not exist or is not managed by this Origin-Server.
409	Conflict	Specified action not appropriate for specified instance.

HTTP/1.1 Code	Reason Phrase	Description
503	Busy	Service is unavailable.

### 6.9.7.3.2 Response Headers

If the CreateSubscriptionrequest was accepted but the deletion lock was not, the response message shall include the following HTTP/1.1 Warning header field:

- Warning: 299 {+SERVICE}: Deletion Lock not granted.

If the request was rejected with an HTTP/1.1 403 status code because Filtered Global Subscription is not supported, the response message shall include the following HTTP/1.1 Warning header field:

- Warning: 299 {+SERVICE}: The Origin-Server does not support Global Subscription Filtering.

### 6.9.7.3.3 Response Message Body

The response message body shall be empty.

## 6.9.8 SuspendGlobalSubscription

This resource suspends an existing Global Subscription or Filtered Global Subscription. The Origin-Server will no longer automatically subscribe the User-Agent to newly-created UPS Instances. This does not delete any existing subscriptions to specific UPS Instances.

### 6.9.8.1 Request

The request message shall be formed as follows:

- Resource
  - {+SERVICE}/workitems/1.2.840.10008.5.1.4.34.5/subscribers/{AETitle}/suspend
  - {+SERVICE}/workitems/1.2.840.10008.5.1.4.34.5.1/subscribers/{AETitle}/suspend

where

  - {+SERVICE} is the base URL for the service. This may be a combination of protocol (either HTTP or HTTPS), authority and path.
  - {AETitle} identifies the subscribed Application Entity.
- Method
  - POST
- The request body shall be empty.

### 6.9.8.2 Behavior

The SuspendGlobalSubscription Origin-Server shall behave as described in Section 6.9.7.2, "Behavior".

### 6.9.8.3 Response

#### 6.9.8.3.1 Response Status Line

The Service shall return an HTTP/1.1 status line, including a status code and associated reason phrase.

If the SuspendGlobalSubscriptionrequest was successful, the Service shall return an HTTP/1.1 "200 - OK" response code.

If the subscription change fails, the Service shall return an appropriate failure status line with a response code from Table 6.9.8-1, "Status Codes".

**Table 6.9.8-1. Status Codes**

HTTP/1.1 Code	Reason Phrase	Description
200	OK	The subscription was suspended.
400	Bad Request	The UPS-RS Origin-Server was unable to understand the request
401	Unauthorized	The UPS-RS Origin-Server refused to accept the request because the client is not authenticated.
403	Forbidden	The UPS-RS Origin-Server understood the request, but is refusing to perform the query (e.g., an authenticated user with insufficient privileges).
404	Not found	The specified UPS Instance or well-known UID does not exist or is not managed by this Origin-Server.
409	Conflict	Specified action not appropriate for specified instance.
503	Busy	Service is unavailable.

### 6.9.8.2.2 Response Message Body

The response message body shall be empty.

## 6.9.9 DeleteSubscription

This resource removes existing subscriptions from the specified UPS Instances.

### 6.9.9.1 Request

The request message shall be formed as follows:

- Resource
  - {+SERVICE}/workitems/{UPSInstanceUID}/subscribers/{AETitle}
 

where

    - {+SERVICE} is the base URL for the service. This may be a combination of protocol (either HTTP or HTTPS), authority and path.
    - {UPSInstanceUID} is the UID of the Unified Procedure Step Instance or a well-known UID.
    - {AETitle} identifies the subscribed Application Entity.
- Method
  - DELETE
- The request body shall be empty.

### 6.9.9.2 Behavior

The DeleteSubscription Origin-Server shall behave as described in Section 6.9.7.2, "Behavior".

### 6.9.9.3 Response

#### 6.9.9.3.1 Response Status Line

The Service shall return an HTTP/1.1 status line, including a status code and associated reason phrase.

If the DeleteSubscriptionrequest was successful, the Service shall return an HTTP/1.1 "200 - OK" response code.

If the subscription fails, the Service shall return an appropriate failure status line with a response code from Table 6.9.7-1, "Status Codes".

**Table 6.9.7-1. Status Codes**

HTTP/1.1 Code	Reason Phrase	Description
200	OK	The subscription was removed.
400	Bad Request	The UPS-RS Origin-Server was unable to understand the request
401	Unauthorized	The UPS-RS Origin-Server refused to accept the request because the client is not authenticated.
403	Forbidden	The UPS-RS Origin-Server understood the request, but is refusing to perform the query (e.g., an authenticated user with insufficient privileges).
404	Not found	The specified UPS Instance or well-known UID does not exist or is not managed by this Origin-Server.
409	Conflict	Specified action not appropriate for specified instance.
503	Busy	Service is unavailable.

### 6.9.9.3.2 Response Message Body

The response message body shall be empty.

## 6.9.10 OpenEventChannel

This resource opens a WebSocket channel that will be used to send Event Reports to the client.

See RFC-6455 for details on the WebSocket protocol.

### 6.9.10.1 Request

The request message shall be formed as follows:

- Resource
  - {+WSSERVICE}/subscribers/{AETitle}

where

  - {+WSSERVICE} is the base URL for the WebSocket service. This shall include the WebSocket protocol (either WS or WSS) and may include a combination of authority and path
  - {AETitle} identifies the subscribed Application Entity.
- Method
  - GET

### 6.9.10.2 Behavior

The Origin-Server maintains the active WebSocket connection and uses it to send Event Report messages for UPS Instances which have subscriptions association with {AETitle} (see Section 6.9.7.2, "Behavior").

If the WebSocket connection is lost at any point the User-Agent can re-establish it by repeating the request.

The state of a WebSocket connection does not affect subscriptions and an Origin-Server is not required to queue messages when the connection is down.

**Note**

A User-Agent will only receive the initial state of a newly-subscribed UPS Instance if the WebSocket connection was initiated before creating the subscription

**6.9.10.3 Response****6.9.10.3.1 Response Status Line**

The Service shall return an HTTP/1.1 status line, including a status code and associated reason phrase.

If the request was successful, the Service shall return an HTTP/1.1 "101 - Switching Protocols" response code.

If the request fails, the Service shall return an appropriate failure status line with a response code from Table 6.9.10-1, "Status Codes".

**Table 6.9.10-1. Status Codes**

HTTP/1.1 Code	Reason Phrase	Description
101	Switching Protocols	The WebSocket connection was established.
400	Bad Request	The UPS-RS Origin-Server was unable to understand the request
401	Unauthorized	The UPS-RS Origin-Server refused to accept the request because the client is not authenticated.
403	Forbidden	The UPS-RS Origin-Server understood the request, but is refusing to perform the query (e.g., an authenticated user with insufficient privileges).
503	Busy	Service is unavailable.

**6.9.10.3.2 Response Message Body**

The response message body shall be empty.

The connection remains open and may be used by the server to send Event messages (see Section 6.9.11, "SendEventReport").

**6.9.11 SendEventReport**

This operation sends an Event Report over an established WebSocket connection.

**6.9.11.1 Request**

The request message shall be formed as follows:

- Resource
  - N/A
- Method
  - WebSocket Data Frame transmission
- The Event Report shall contain all mandatory attributes described in Table CC.2.4-1 in PS3.4 and Table 10.3-1 in PS3.7 for the event type.

**6.9.11.1.1 Request Message Body**

WebSocket Events are encoded as WebSocket data frames with an opcode of "%x1" (text).

The frame payload data shall be a DICOM JSON dataset containing the attributes of the Event Report.

---

**Note****1. Example WebSocket payload:**

```
{
  "00000002": [ "1.2.840.10008.5.1.4.34.6.4" ],
  "00000100": [ 256 ],
  "00000110": [ 23 ],
  "00001000": [ "1.2.840.10008.5.1.4.34.6.4.2.3.44.22231" ],
  "00001001": [ 1 ],
  "00741238": [ "SCHEDULED" ],
  "00744041": [ "READY" ]
}
```

2. The WebSocket protocol does not allow content negotiation so it is not possible to support both XML and JSON encoding of Event Report messages without extending the protocol.

### **6.9.11.2 Behavior**

Section CC.2.4.3 in PS3.4 describes the scenarios in which an Origin-Server sends Event Reports to a subscriber and the content of the Event Report messages.

### **6.9.11.3 Response**

None.



# 7 Persistent Object Types

The provisions for some specific object types shall be as defined in this section.

## Note

In all cases the categorization depends on the SOP Class of the objects, enabling a client, or application building an HTML page for the client, to determine in advance of the request what the requirements will be.

## 7.1 Single Frame Image Objects

### 7.1.1 Objects Accessed

In this category are all object instances of SOP classes defined in PS3.3 that consist of a single image frame, instances of multi-frame SOP Classes that contain only one frame, or object instances that consist of single frame accessed from instances of multi-frame SOP Classes using the "frameNumber" parameter.

### 7.1.2 MIME Type Constraints

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

- WADO-URI and WADO-WS
  - application/dicom
  - image/jpeg
- WADO-RS
  - application/dicom
  - application/octet-stream
  - application/dicom+xml
  - application/json

If the contentType parameter is not present in the URI or WADO-WS request, the response shall contain an image/jpeg MIME type, if compatible with the 'Accept' field of the GET method. If the contentType parameter is not present in the WADO-RS request, the response is dependent on the 'Accept' field and the requested resource.

When an image/jpeg MIME type is returned, the image shall be encoded using the JPEG baseline lossy 8 bit Huffman encoded non-hierarchical non-sequential process ISO/IEC 10918.

## Note

The choice of image/jpeg as the default for continuous tone images is a consequence of the universal support by Web Clients.

The Server should also support the following MIME types for WADO-URI or WADO-WS:

- image/gif
- image/png
- image/jp2

The Server should also support the following MIME types for WADO-RS:

- image/dicom
- image/dicom+jpeg

- image/dicom+rl
- image/dicom+jpeg-ls
- image/dicom+jp2image/dicom+jpx

The Server may also support other MIME types.

## 7.2 Multi-frame and Video Image Objects

### 7.2.1 Objects Included

In this category are all SOP classes defined in PS3.3 that are multi-frame or video image objects.

### 7.2.2 MIME Type Constraints

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

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

If the contentType parameter is not present in the WADO-URI or WADO-WS request, the response shall contain a application/dicom MIME type.

The Server can optionally support the following MIME types for WADO-URI and WADO-WS:

- video/mpeg
- image/gif

The Server can optionally support the following MIME types for WADO-RS:

- image/dicom+jpx
- video/mpeg
- video/mp4

The Server may also support other MIME types.

## 7.3 Text Objects

### 7.3.1 Objects Included

In this category are all SOP classes defined in PS3.3 that include the SR Document Content Module.

#### Note

This includes all SOP Classes that are SR documents, such as narrative text, structured reports, CAD, measurement reports and key object selection documents.

### 7.3.2 MIME Type Constraints

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

- application/dicom
- text/plain
- text/html

If the contentType parameter is not present in the request, or contains only MIME types that the Server does not support, the response shall contain a text/html MIME type.

It is recommended that the Server also support the following MIME types:

- text/xml
- application/pdf
- text/rtf
- a "CDA" MIME type, in conformance to HL7 CDA R2, e.g., text/xml

The Server may also support other MIME types.

## 7.4 Other Objects

### 7.4.1 Objects Included

The category shall include all objects of all SOP classes defined in PS3.3 that are not included in the categories described in the sections above, and that are considered in PS3.3 as classes of persistent objects.

### 7.4.2 MIME Type Constraints

The Server shall be able to send a response in the following MIME type:

- application/dicom

The Server may also support other MIME types.

If the contentType parameter is not present in the request, the response shall contain an application/dicom MIME type.



# 8 Parameters of the Request

## 8.1 Parameters Available for all DICOM Persistent Objects

Parameters specified in this section are applicable to all supported DICOM SOP Classes.

### Note

To identify a DICOM Object, only one UID is required, because any UID is globally unique. However, the standard requires that the UID of the higher levels in the DICOM Information Model are specified (i.e., series and study), in order to support the use of DICOM devices that support only the baseline hierarchical (rather than extended relational) Query/Retrieve model, which requires the Study Instance UID and Series Instance UID to be defined when retrieving an SOP Instance, as defined in PS3.4.

### 8.1.1 Request Type

Type of request performed. This parameter is REQUIRED for URI based mode.

The parameter name shall be "requestType".

The value shall be "WADO".

### Note

This parameter allows other types of requests to be introduced in the future, using a similar syntax.

### 8.1.2 Unique Identifier of the Study

Study Instance UID as defined in PS3.3. This parameter is REQUIRED.

The parameter name shall be "studyUID" for URI based mode, and "StudyRequest" that contains a required "studyInstanceUID" attribute for the WS mode.

The value shall be encoded as a Unique Identifier (UID) string, as specified in PS3.5, except that it shall not be padded to an even length with a NULL character.

### 8.1.3 Unique Identifier of the Series

Series Instance UID as defined in PS3.3. This parameter is REQUIRED.

The parameter name shall be "seriesUID" for URI based mode, and, for the WS mode, one or multiple "SeriesRequest" that is included into the above described "StudyRequest" and that contains a required "seriesInstanceUID" attribute.

The value shall be encoded as a Unique Identifier (UID) string, as specified in PS3.5, except that it shall not be padded to an even length with a NULL character.

### 8.1.4 Unique Identifier of the Object

SOP Instance UID as defined in PS3.3. This parameter is REQUIRED.

The parameter name shall be "objectUID" for URI based mode, and for the WS mode one or multiple "DocumentRequest" that is included into the above described "SeriesRequest" and that include each one:

- a required "DocumentUniqueid" that contains the Instance UID,
- an optional "RepositoryUniqueid" that contains the UID of the DICOM server, and
- an optional "HomeCommunityId" that contains the UID of the "clinical affinity domain".

The value shall be encoded as a unique identifier (UID) string, as specified in PS3.5, except that it shall not be padded to an even length with a NULL character.

### 8.1.5 MIME Type of the Response

MIME type(s) desired by the Web Client for the response from the Server, as defined in the IETF RFC7230. This parameter is OPTIONAL for URI based mode, it shall be present for the WS mode "Rendered Requester" and shall not be present in the other WS mode transactions.

The parameter name shall be "contentType" for URI based mode, and, for the WS mode, "ContentTypeList" that contains one or multiple "ContentType".

In URI based mode, the value shall be a list of MIME types, separated by a "," character, and potentially associated with relative degree of preference, as specified in IETF RFC7230. In WS mode, it contains one or more "ContentType" elements containing each one MIME type.

In URI based mode, the Web Client shall provide list of content types it supports in the "Accept" field of the GET method. The value of the contentType parameter of the request shall be one of the values specified in that field.

#### Note

1. Typically the Accept field will be sent by a Web Client as "\*\*/\*", which is compatible with any MIME types.
2. When this parameter is absent, the default content type of the response is dictated by the "MIME type constraints" sub-sections of Section 7 (i.e., 7.1.2, 7.2.2, 7.3.2, 7.4.2).

### 8.1.6 Charset of the Response

Character set with which the returned objects are to be encoded, as defined in the IETF RFC7230. This parameter is OPTIONAL for URI based mode, and for the WS mode "Rendered Requester" and shall not be present in the other WS mode transactions.

The parameter name shall be "charset" for URI based mode, and "CharsetList" containing one or more elements "Charset" for the WS mode.

For the URI mode, the value shall be a list of character sets, separated by a "," character, and potentially associated with relative degree of preference, as specified in IETF RFC7230.

In URI based mode, the Web Client may provide a list of character sets it supports in the "Accept-charset" field of the GET method. If this field is present, the value of the charset parameter of the request shall be one of the values specified in it.

The Web Server may or may not support character set conversion. If character set conversion is supported:

- text based DICOM objects retrieved other than as application/dicom MIME type (e.g., text/plain) may be returned in the requested character set (converted if necessary)
- DICOM objects retrieved as application/dicom MIME type have all contained strings returned in the requested character set (converted if necessary) and the Specific Character Set (0008,0005) updated (if necessary)

#### Note

1. The IANA Character Set registrations specify names and multiple aliases for most character sets. The standard value for use in WADO is the one marked by IANA as "preferred for MIME." If IANA has not marked one of the aliases as "preferred for MIME", the name used in DICOM shall be the value used for WADO.
2. The table in Annex D provides an informative mapping of some IANA values to DICOM Specific Character Set Defined Terms.

### 8.1.7 Anonymize Object

Removal of all patient identification information from within the DICOM objects, if not already done, as defined in PS3.15. This parameter is OPTIONAL. In the URI based mode, it shall only be present if contentType is application/dicom.

This parameter is Optional

The parameter name shall be "anonymize" for URI based mode, and "Anonymize" for the WS mode.

The value shall be "yes".

The Server may return an error if it either cannot or refuses to anonymize these objects.

In WS mode, the metadata describing the objects or information extracted from them in the response shall be anonymized if requested.

The Server shall return a new SOP Instance UID if the content of the object has not already been anonymized.

#### Note

1. This standard does not introduce any security-related requirements. It is likely that the information contained within DICOM objects identifies the patient. The protocol used (that is HTTP) can be replaced by HTTPS, which is its secure extension, to protect the information in transit. The underlying DICOM implementation decides whether or not to grant access to a particular DICOM object based on whatever security policy or mechanism it has in place. A server is unlikely to fulfill a request from an unknown user (e.g., accessed via the HTTP protocol) unless it is certain that the data requested has no patient identifying information within it and has been approved for public viewing.
2. The Anonymize object enables, for example, teaching files systems or clinical trial applications to offer an access to original images stored in a PACS, without disclosing the patients identity, and requiring storage of a (de-identified) copy of the original image. Anonymization is the responsibility of the Server. In order to preserve patient confidentiality, the Server likely will refuse to deliver an anonymized SOP instance to an unknown or unauthorized person unless the Server is certain that the SOP instance holds no patient identifying information. This would include "blanking out" any annotation area(s) containing nominative information burned into the pixels or in the overlays.

## 8.1.9 Retrieve Partial Information From Objects

Retrieval of additional information from the DICOM objects, using a filtering mechanism based on the XML mapping of DICOM IODs, as described in the Native DICOM Model defined in PS3.19. This parameter is defined only for the WS mode "Information Requester" transaction.

The parameter name shall be "XPath".

## 8.2 Parameters for DICOM Image Persistent Objects

These parameters shall only be included when a request is made for a Single Frame Image Objects or Multi-Frame Image or video Objects as defined in Section 7.2.

### 8.2.1 Annotation On The Object

Annotation of objects retrieved and displayed as an image. This parameter is OPTIONAL for the URI based mode and the WS mode "Rendered Requester" transaction. It shall not be present if contentType is application/dicom, or is a non-image MIME type (e.g., text/\*). When it is not present for image objects, no additional annotation may be burnt in.

When used in conjunction with a presentation state object, it shall be applied after the presentation on the images. When used in conjunction with the region parameter, it shall be applied after the selection of the region.

The parameter name shall be "annotation" for URI based mode, and "Annotation" for the WS mode. Its value is a non-empty list of one or more of the following items, separated by a "," character:

- "patient", for displaying patient information on the image (e.g., patient name, birth date,...)
- "technique", for displaying technique information of the image (e.g., image number, study date, image position,...).

#### Note

The exact nature and presentation of the annotation is determined by the Server. The annotation is burned into the returned image pixels.

### 8.2.2 Number of Pixel Rows

The parameter name shall be "rows" for URI based mode, and "Rows" for the WS mode.

The value shall be expressed as an integer, representing the image height to be returned. It is OPTIONAL for the URI based mode and the WS mode "Rendered Requester" transaction. It shall not be present for other WS mode transactions. It shall not be present if contentType is application/dicom.

If both "rows" and "columns" are specified, then each shall be interpreted as a maximum, and a size will be chosen for the images within these constraints, maintaining the correct aspect ratio. If the number of rows is absent and the number of columns is present, the number of rows shall be chosen in order to maintain the correct aspect ratio. If both are absent, the images (or selected region) are sent in their original size (or the size of the presentation state applied on the images), resulting as one pixel of screen image for each value in the images data matrix.

The value shall be encoded as an integer string (IS), as specified in PS3.5.

### 8.2.3 Number of Pixel Columns

The parameter name shall be "columns" for URI based mode, and "Columns" for the WS mode.

The value shall be expressed as an integer, representing the image width to be returned. It is OPTIONAL for the URI based mode and the WS mode "Rendered Requester" transaction. It shall not be present if contentType is application/dicom.

If both "rows" and "columns" are specified, then each shall be interpreted as a maximum, and a size will be chosen for the images within these constraints, maintaining the correct aspect ratio. If the number of columns is absent and the number of rows is present, the number of columns shall be chosen in order to maintain the correct aspect ratio. If both are absent, the images (or selected region) is sent in its original size (or the size of the presentation state applied on the images), resulting as one pixel of screen for one pixel of the images.

The value shall be encoded as an integer string (IS), as specified in PS3.5.

### 8.2.4 Region of the Image

This parameter allows selection of a rectangular region of an image matrix to be retrieved. The purpose of this parameter is to allow a user to view a selected area of the image matrix, for example at higher magnification.

The parameter is OPTIONAL for the URI based mode and the WS mode "Rendered Requester" transaction. It shall not be present for other WS mode transactions.

The parameter name shall be "region" for URI based mode, and "Region" for the WS mode.

It shall not be present if contentType is application/dicom.

The value shall be expressed as a list of four positive decimal strings, separated by the ',' character, representing the region of the source images to be returned. These decimal values shall be values in a normalized coordinate system relative to the size of the original image matrix measured in rows and columns, with values ranging from 0.0 to 1.0, and representing in the following order:

- the x position of the top left hand corner of the region to be retrieved, 0.0 corresponding to the first column of the image matrix. In the WS mode, this value is encoded into an XML element "XMin".
- the y position of the top left hand corner of the region to be retrieved, 0.0 corresponding to the top row of the image matrix. In the WS mode, this value is encoded into an XML element "YMin".
- the x position of the bottom right hand extent of the region, 1.0 corresponding to the last column of the image matrix, 0.0 being forbidden. In the WS mode, this value is encoded into an XML element "XMax".
- the y position of the bottom right hand extent of the region, 1.0 corresponding to the last row of the image matrix, 0.0 being forbidden. In the WS mode, this value is encoded into an XML element "YMax".

#### Note

The Server may or may not support this parameter.

If this parameter is supported, an image matrix corresponding to the specified region shall be returned with size corresponding to the specified normalized coordinate values otherwise the complete image matrix shall be returned. If the presentationUID parameter is present, the region shall be selected after the corresponding presentation state has been applied on the images.

### 8.2.5 Window Center of the Image

The parameter name shall be "windowCenter" for URI based mode, and "WindowCenter" for the WS mode.

Controls the luminosity of the images as defined in PS3.3. This parameter is OPTIONAL for the URI based mode and the WS mode "Rendered Requester" transaction. It shall not be present for other WS mode transactions. This parameter is REQUIRED if "windowWidth" or "WindowWidth" is present. This parameter shall not be present if there is a presentationUID parameter. It shall not be present if contentType is application/dicom.

The value shall be encoded as a decimal string (DS), as specified in PS3.5.

### 8.2.6 Window Width of the Image

The parameter name shall be "windowWidth" for URI based mode, and "WindowWidth" for the WS mode.

Controls the contrast of the images as defined in PS3.3. This parameter is OPTIONAL for the URI based mode and the WS mode "Rendered Requester" transaction. It shall not be present for other WS mode transactions. It is REQUIRED if "windowCenter" or "WindowCenter" is present. This parameter shall not be present if there is a presentationUID parameter. It shall not be present if contentType is application/dicom.

The value shall be encoded as a decimal string (DS), as specified in PS3.5.

### 8.2.7 Frame Number

The parameter name shall be "frameNumber" for URI based mode, and "FrameNumber" for the WS mode.

Specifies that the single frame with that number within a multi-frame image object, as defined in PS3.3 that shall be returned. It is OPTIONAL and shall be ignored in the case of all objects other than multi-frame objects. It shall not be present if contentType is application/dicom.

The value shall be encoded as an integer string (IS), as specified in PS3.5.

### 8.2.8 Image Quality

The parameter name shall be "imageQuality" for URI based mode, and "ImageQuality" for the WS mode. It is OPTIONAL for the URI based mode and the WS mode "DICOM requester" and "Rendered Requester" transactions. It shall not be present if contentType is application/dicom, except if the transferSyntax parameter is present and corresponds to a lossy compression.

If the requested MIME type is for a lossy compressed image (e.g., image/jpeg), this parameter indicates the required quality of the image to be returned within the range 1 to 100, 100 being the best quality.

#### Note

Decompression and re-compression may degrade the image quality if the original image was already irreversibly compressed. In case the image has been already lossy compressed using the same format as required (e.g., jpeg), it may be sent as it is without decompressing and re-compressing it.

The value shall be encoded as an integer string (IS), as specified in PS3.5.

#### Note

The specific interpretation of the meaning of this parameter is left to the interpretation of the implementers of the standard.

### 8.2.9 Unique Identifier of the Presentation Object

The parameter name shall be "presentationUID" for URI based mode, and "PresentationUID" for the WS mode.

SOP Instance UID of the presentation state storage object to be applied to the images. This parameter is OPTIONAL for the URI based mode and the WS mode "Rendered Requester" transaction. It shall not be present if contentType is application/dicom.

The value shall be encoded as a unique identifier (UID) string, as specified in PS3.5, except that it shall not be padded to an even length with a NULL character.

If this parameter is combined with region and/or annotation parameter(s), the presentation state shall be applied to the images prior to selecting a region and burning in annotations.

If the Presentation Size Mode in the presentation state is SCALE TO FIT or TRUE SIZE, then the displayed area specified in the presentation shall be scaled to fit the size specified by the rows and columns parameters if present, otherwise the displayed area selected in the presentation state will be returned without scaling.

**Note**

1. The intent of the TRUE SIZE mode in the presentation state cannot be satisfied, since the physical size of the pixels displayed by the web browser is unlikely to be known. If the Presentation Size Mode in the presentation state is MAGNIFY, then the displayed area specified in the presentation shall be magnified (scaled) as specified in the presentation state. It will then be cropped to fit the size specified by the rows and columns parameters, if present.
2. Any Displayed Area relative annotations specified in the presentation state are rendered relative to the Specified Displayed Area within the presentation state, not the size of the returned image.

Though the output of the presentation state is defined in DICOM to be in P-Values (grayscale values intended for display on a device calibrated to the DICOM Grayscale Standard Display Function PS3.14), the grayscale or color space for the images returned by the request is not defined by this standard.

## **8.2.10 Unique Identifier of the Series Containing The Presentation Object**

The parameter name shall be "presentationSeriesUID" for URI based mode, and "PresentationSeriesUID" for the WS mode.

Series Instance UID of the series containing the presentation state storage object to be applied on the images. This parameter is REQUIRED and shall only be present if "presentationUID" is present.

The value shall be encoded as a unique identifier (UID) string, as specified in PS3.5, except that it shall not be padded to an even length with a NULL character.

**Note**

As specified in DICOM, the Presentation State will be in the same study as the images it applies to.

## **8.2.11 Transfer Syntax UID**

The parameter name shall be "transferSyntax" for URI based mode, and "TransferSyntaxUIDList" containing one or more "TransferSyntaxUID" elements for the WS mode.

The Transfer Syntax to be used within the DICOM image objects, as specified in PS3.6. This parameter is OPTIONAL for the URI based mode and the WS mode "DICOM Requester" transaction. It shall not be present if contentType is other than application/dicom.

By default the DICOM object(s) returned shall be encoded in Explicit VR Little Endian. Neither Implicit VR, nor Big Endian shall be used. The response shall be the Transfer Syntax requested if possible. If it is not possible for the response to be sent using the requested transfer syntax then the Explicit VR Little Endian Uncompressed Transfer Syntax shall be used.

**Note**

The transfer syntax can be chosen as one of the values of TransferSyntaxUID corresponding to JPIP, in case of which the returned objects will contain the URL of the JPIP session to launch for retrieving the corresponding image.

The value shall be encoded as an unique identifier (UID) string, as specified in PS3.5, except that it shall not be padded to an even length with a NULL character.

# A URI Query Component Syntax (Normative)

This Standard uses the URI syntax as defined in IETF RFC 3986 *Uniform Resource Identifier (URI): Generic Syntax* and extends it by specifying the syntax of the query component of DICOM URIs. The grammar for the query component is defined using IETF RFC 5234 *Augmented BNF for Syntax Specifications: ABNF*.

DICOM URIs may use the query component of the URI to specify request parameters. The following grammar defines the general syntax of parameters contained in the query component of the URI. Specific HTTP transactions defined elsewhere in this standard may further refine the legal <name> and/or <value> rules.

```

query-component = parameter [ *("&" parameter) ]
parameter      = name "=" value
name           = *qchar
value          = *qchar
qchar          = unreserved / pct-encoded / qspecial
qspecial       = "/" / "?" / "." / "@" / "!" / "$" / "'"
               / "(" / ")" / "*" / "+" / "," / ";"

```

The following rules are defined in IETF RFC3986 (Normative). They are reproduced here for convenience.

```

unreserved  = ALPHA / DIGIT / "-" / "." / "_" / "~"
pct-encoded = "%" HEXDIG HEXDIG

```

## Note

1. This grammar allows the query component to contain any of the legal characters as defined by RFC 3986.
2. No whitespace is permitted in URIs. Whitespace around line breaks and the line breaks themselves should be stripped before parsing the URI (See RFC 3986 Appendix C).
3. RFC 3986 does not permit an empty query component, i.e. if the "?" appears in the URI then there must be some legal query parameters in the URI.
4. The <qchar> rule defined above is the <pchar> rule of RFC 3986, which defines the legal character for the query component, minus the characters "=" and "&".



# B Examples (Informative)

## B.1 Retrieving a Simple DICOM Image in JPEG

```
http://www.hospital-stmarco/radiology/wado.php?requestType=WADO
&studyUID=1.2.250.1.59.40211.12345678.678910
&seriesUID=1.2.250.1.59.40211.789001276.14556172.67789
&objectUID=1.2.250.1.59.40211.2678810.87991027.899772.2
```

## B.2 Retrieving a DICOM SR in HTML

```
http://server234/script678.asp?requestType=WADO
&studyUID=1.2.250.1.59.40211.12345678.678910
&seriesUID=1.2.250.1.59.40211.789001276.14556172.67789
&objectUID=1.2.250.1.59.40211.2678810.87991027.899772.2
&charset=UTF-8
```

## B.3 Retrieving a Region of A DICOM Image

Retrieving a region of a DICOM image, converted if possible in JPEG2000, with annotations burned into the image containing the patient name and technical information, and mapped into a defined image size:

```
https://aspradio/imageaccess.js?requestType=WADO
&studyUID=1.2.250.1.59.40211.12345678.678910
&seriesUID=1.2.250.1.59.40211.789001276.14556172.67789
&objectUID=1.2.250.1.59.40211.2678810.87991027.899772.2
&contentType=image%2Fjp2;level=1,image%2Fjpeg;q=0.5
&annotation=patient,technique
&columns=400
&rows=300
&region=0.3,0.4,0.5,0.5
&windowCenter=-1000
&windowWidth=2500
```

## B.4 Retrieving As A DICOM MIME Type

Retrieving a DICOM image object using the baseline 8-bit lossy JPEG transfer syntax, and de-identified:

```
http://www.medical-webservice.st/RetrieveDocument?requestType=WADO
&studyUID=1.2.250.1.59.40211.12345678.678910
&seriesUID=1.2.250.1.59.40211.789001276.14556172.67789
&objectUID=1.2.250.1.59.40211.2678810.87991027.899772.2
&contentType=application%2Fdicom
&anonymize=yes
&transferSyntax=1.2.840.10008.1.2.4.50
```



# C Applications (Informative)

There are multiple applications, in which DICOM and "web-based" environments are interacting. "Web-based" means information and communication systems that are using Internet related technologies (Web, e-mail...). The basic feature supported by this standard is a mechanism for the "Web-based" system to retrieve a DICOM persistent object from the "DICOM-based" system.

Typical applications are:

- i. Referencing an image or a report from an electronic patient record (EPR)
- ii. Including references to images in an e-mail
- iii. Providing access by outside referring doctors to a hospital web server that contains references to reports, images and waveforms
- iv. Providing access to anonymized DICOM reports, images and waveforms via a web server, for teaching purposes and for clinical trials.

To retrieve DICOM persistent objects using "WADO", the "web-based" system must "know" the UIDs (Study, Series, SOP Instance) of the objects it needs to retrieve. These may be obtained through different methods (reception of a standardized message containing a document containing the reference to the DICOM objects, query of other systems...) that are beyond the scope of this standard.



# D IANA Mapping (Informative)

The following table provides an informative mapping of some IANA values to DICOM Specific Character Set Defined Terms:

**Table D-1. IANA Mapping**

IANA	DICOM	Character Set
ISO-8859-1	ISO_IR 100	Latin alphabet #1
ISO-8859-2	ISO_IR 101	Latin alphabet #2
ISO-8859-3	ISO_IR 109	Latin alphabet #3
ISO-8859-4	ISO_IR 110	Latin alphabet #4
ISO-8859-5	ISO_IR 144	Cyrillic
ISO-8859-6	ISO_IR 127	Arabic
ISO-8859-7	ISO_IR 126	Greek
ISO-8859-8	ISO_IR 138	Hebrew
ISO-8859-9	ISO_IR 148	Latin alphabet #5
TIS-620	ISO_IR 166	Thai
ISO-2022-JP	ISO 2022 IR 87	Japanese
ISO-2022-KR	ISO 2022 IR 149	Korean
ISO-2022-CN	ISO 2022 IR 58	Chinese
GB18030	GB18030	Chinese
GBK	GBK	Chinese
UTF-8	ISO_IR 192	Unicode



# E WADO WS Schemas and Examples

## E.1 WADO WS XSD Schema (Informative)

The following XSD schema can be used for the WADO WS implementation:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="urn:ihe:rad:xdsi-b:2009"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:ihe:rad:xdsi-b:2009" elementFormDefault="qualified"
attributeFormDefault="unqualified"
xmlns:tns="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0">
  <xs:import namespace="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0" />
  <xs:import namespace="urn:ihe:iti:xds-b:2007" />
  <xs:simpleType name="LongName">
    <xs:restriction base="xs:string">
      <xs:maxLength value="256" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="RetrieveDocumentSetRequestType">
    <xs:sequence>
      <xs:element name="DocumentRequest" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="HomeCommunityId" type="LongName" minOccurs="0">
              <xs:annotation>
                <xs:documentation>This corresponds to the home attribute of the
                Identifiable class in regrep RIM (regrep-rim-3.0-os.pdf, page
                20)</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="RepositoryUniqueId" type="LongName"
            minOccurs="0">
              <xs:annotation>
                <xs:documentation>This is the
                XDSDocumentEntry.repositoryUniqueId attribute in the XDS
                metadata</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="DocumentUniqueId" type="LongName">
              <xs:annotation>
                <xs:documentation>This is the XDSDocumentEntry.uniqueId
                attribute in the XDS metadata</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RegistryErrorType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="codeContext" type="xs:string" use="required" />
        <xs:attribute name="errorCode" type="xs:string" use="required" />
        <xs:attribute default="urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error"
        name="severity" type="xs:anyURI" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

```

    <xs:attribute name="location" type="xs:string" use="optional" />
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="RegistryErrorListType">
  <xs:annotation>
    <xs:documentation xml:lang="en">The RegistryErrorList is derived from the
      ErrorList element from the ebXML Message Service
      Specification</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element maxOccurs="unbounded" name="RegistryError"
      type="RegistryErrorType" />
  </xs:sequence>
  <xs:attribute name="highestSeverity" type="xs:anyURI" use="optional" />
</xs:complexType>
<xs:complexType name="RegistryResponseType">
  <xs:annotation>
    <xs:documentation xml:lang="en">Base type for all ebXML Registry
      responses</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <!-- every response may be extended using Slots. -->
    <xs:element minOccurs="0" type="RegistryErrorListType"
      name="RegistryErrorList" />
  </xs:sequence>
  <xs:attribute name="status" type="xs:anyURI" use="required" />
  <!-- id is the request id for the request for which this is a response -->
  <xs:attribute name="requestId" type="xs:anyURI" use="optional" />
</xs:complexType>
<xs:element name="RetrieveDocumentSetRequest"
  type="RetrieveDocumentSetRequestType" />
<xs:complexType name="RetrieveDocumentSetResponseType">
  <xs:sequence>
    <xs:element name="RegistryResponse" type="RegistryResponseType" />
    <xs:sequence minOccurs="0">
      <xs:element maxOccurs="unbounded" name="DocumentResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="HomeCommunityId" type="LongName">
              <xs:annotation>
                <xs:documentation>This corresponds to the home attribute of
                  the Identifiable class in regrep RIM (regrep-rim-3.0-os.pdf,
                  page 20)</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="RepositoryUniqueId" type="LongName">
              <xs:annotation>
                <xs:documentation>This is the
                  XDSDocumentEntry.repositoryUniqueId attribute in the XDS
                  metadata</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="DocumentUniqueId" type="LongName">
              <xs:annotation>
                <xs:documentation>This is the XDSDocumentEntry.uniqueId
                  attribute in the XDS metadata</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:sequence>
  <xs:element minOccurs="0" name="Document">

```

```

        type="xs:base64Binary" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:sequence>
</xs:complexType>
<xs:element name="RetrieveDocumentSetResponse"
type="RetrieveDocumentSetResponseType" />
<xs:complexType name="RetrieveRenderedImagingDocumentSetRequestType">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" name="StudyRequest">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" name="SeriesRequest">
            <xs:complexType>
              <xs:sequence>
                <xs:element maxOccurs="unbounded"
name="RenderedDocumentRequest">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="HomeCommunityId" type="LongName"
minOccurs="0">
                        <xs:annotation>
                          <xs:documentation>This corresponds to the home
attribute of the Identifiable class in regrep RIM
(regrep-rim-3.0-os.pdf, page 20)</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                      <xs:element name="RepositoryUniqueld" type="LongName"
minOccurs="0">
                        <xs:annotation>
                          <xs:documentation>This is the
XSDSDocumentEntry.repositoryUniqueld attribute in
the XDS metadata</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                      <xs:element name="DocumentUniqueld" type="LongName">
                        <xs:annotation>
                          <xs:documentation>This is the
XSDSDocumentEntry.uniqueld attribute in the XDS
metadata</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                      <xs:element minOccurs="0" name="Annotation"
type="xs:string" />
                      <xs:element minOccurs="0" name="Rows"
type="xs:string" />
                      <xs:element minOccurs="0" name="Columns"
type="xs:string" />
                      <xs:element minOccurs="0" name="Region"
type="xs:string" />
                      <xs:element minOccurs="0" name="WindowWidth"
type="xs:string" />
                      <xs:element minOccurs="0" name="WindowCenter"
type="xs:string" />
                      <xs:element minOccurs="0" name="ImageQuality"
type="xs:string" />
                      <xs:element minOccurs="0" name="PresentationSeriesUID"
type="xs:string" />

```

```

    <xs:element minOccurs="0" name="Anonymize"
      type="xs:string" />
    <xs:element minOccurs="0" name="FrameNumber"
      type="xs:string" />
    <xs:element minOccurs="1" name="ContentTypeList"
      type="xs:string" maxOccurs="unbounded" />
    <xs:element minOccurs="1" name="CharsetList"
      type="xs:string" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:element name="RetrieveRenderedImagingDocumentSetRequest"
  type="RetrieveRenderedImagingDocumentSetRequestType" />
<xs:complexType name="RetrieveRenderedImagingDocumentSetResponseType">
  <xs:sequence>
    <xs:element name="RegistryResponse" type="RegistryResponseType" />
    <xs:element maxOccurs="unbounded" name="RenderedDocumentSetResponse"
      minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="HomeCommunityId" type="LongName" minOccurs="0">
            <xs:annotation>
              <xs:documentation>This corresponds to the home attribute of the
                Identifiable class in regrep RIM (regrep-rim-3.0-os.pdf, page
                20)</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="RepositoryUniqueId" type="LongName"
            minOccurs="1">
            <xs:annotation>
              <xs:documentation>This is the
                XDSDocumentEntry.repositoryUniqueId attribute in the XDS
                metadata</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="SourceDocumentUniqueId" type="LongName">
            <xs:annotation>
              <xs:documentation>This is the XDSDocumentEntry.uniqueId
                attribute in the XDS metadata</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element minOccurs="1" name="Annotation" type="xs:string" />
          <xs:element minOccurs="1" name="Rows" type="xs:string" />
          <xs:element minOccurs="1" name="Columns" type="xs:string" />
          <xs:element minOccurs="1" name="Region" type="xs:string" />
          <xs:element name="WindowWidth" type="xs:string" />
          <xs:element minOccurs="1" name="WindowCenter" type="xs:string" />
          <xs:element minOccurs="1" name="ImageQuality" type="xs:string" />
          <xs:element minOccurs="1" name="PresentationSeriesUID"
            type="xs:string" />
          <xs:element minOccurs="0" name="Anonymize" type="xs:string" />
          <xs:element minOccurs="0" name="FrameNumber" type="xs:string" />

```

```

    <xs:element minOccurs="1" name="mimeType" type="xs:string"
      maxOccurs="1" />
    <xs:element minOccurs="1" name="Document" type="xs:base64Binary" />
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:element name="RetrieveRenderedImagingDocumentSetResponse"
  type="RetrieveRenderedImagingDocumentSetResponseType" />
<xs:complexType name="RetrievalImagingDocumentSetInformationRequestType">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" name="StudyRequest">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" name="SeriesRequest">
            <xs:complexType>
              <xs:sequence>
                <xs:element maxOccurs="unbounded"
                  name="DocumentInformationRequest">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="HomeCommunityId" type="LongName"
                        minOccurs="0">
                        <xs:annotation>
                          <xs:documentation>This corresponds to the home
                            attribute of the Identifiable class in regrep RIM
                            (regrep-rim-3.0-os.pdf, page 20)</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                      <xs:element name="RepositoryUniqueld" type="LongName"
                        minOccurs="1">
                        <xs:annotation>
                          <xs:documentation>This is the
                            XSDDocumentEntry.repositoryUniqueld attribute in
                            the XDS metadata</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                      <xs:element name="DocumentUniqueld" type="LongName">
                        <xs:annotation>
                          <xs:documentation>This is the
                            XSDDocumentEntry.uniqueld attribute in the XDS
                            metadata</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                      <xs:element minOccurs="0" name="Anonymize"
                        type="xs:string" />
                      <xs:element minOccurs="1" name="XPath" type="xs:string"
                        maxOccurs="1" />
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

```

<xs:element name="RetrievalImagingDocumentSetInformationRequest"
type="RetrievalImagingDocumentSetInformationRequestType" />
<xs:complexType name="RetrievalImagingDocumentSetInformationResponseType">
  <xs:sequence>
    <xs:element name="RegistryResponse" type="RegistryResponseType" />
    <xs:element maxOccurs="unbounded" name="DocumentInformationResponse"
minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="HomeCommunityId" type="LongName" minOccurs="0">
            <xs:annotation>
              <xs:documentation>This corresponds to the home attribute of the
Identifiable class in regrep RIM (regrep-rim-3.0-os.pdf, page
20)</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="RepositoryUniqueId" type="LongName"
minOccurs="1">
            <xs:annotation>
              <xs:documentation>This is the
XDSDocumentEntry.repositoryUniqueId attribute in the XDS
metadata</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="DocumentUniqueId" type="LongName">
            <xs:annotation>
              <xs:documentation>This is the XDSDocumentEntry.uniqueId
attribute in the XDS metadata</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element minOccurs="0" name="FrameNumber" type="xs:string" />
          <xs:element minOccurs="1" name="XPathResponseList">
            <xs:complexType>
              <xs:sequence>
                <xs:element maxOccurs="unbounded" name="XPathResponse"
type="xs:string" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:element name="RetrievalImagingDocumentSetInformationResponse"
type="RetrievalImagingDocumentSetInformationResponseType" />
</xs:schema>

```

## E.2 WADO WS Request Example (Informative)

POST /tf6/services/xdsrepositoryb HTTP/1.1

Content-Type: multipart/related; boundary=MIMEBoundaryurn\_uuid\_DCD262C64C22DB97351256303951323;

type="application/xop+xml"; start="<0.urn:uuid:DCD262C64C22DB97351256303951324@apache.org>";

start-info="application/soap+xml";

action="urn:dicom:ws:wado:2011:RetrieveRenderedImagingDocumentSet"

User-Agent: Axis2

Host: localhost:5000

```
--MIMEBoundaryurn_uuid_DCD262C64C22DB97351256303951323
Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <0.urn:uuid:DCD262C64C22DB97351256303951324@apache.org>
<s:Envelope
  xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      urn:dicom:ws:wado:2011:RetrieveRenderedImagingDocumentSet</a:Action>
    <a:MessageID>urn:uuid:0fbfdced-6c01-4d09-a110-2201afedaa02</a:MessageID>
    <a:ReplyTo s:mustUnderstand="1">
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To >http://localhost:2647/XdsService/DocSource.svc</a:To>
  </s:Header>
  <s:Body>
    <RetrieveImagingDocumentSetRequest xmlns:iherad="urn:ihe:rad:xdsi-b:2009"
      xmlns:ihe="urn:ihe:iti:xds-b:2007">
      <StudyRequest studyInstanceUID="1.3.6.1.4...101">
        <SeriesRequest seriesInstanceUID="1.3.6.1.4...201">
          <ihe:DocumentRequest>
            <ihe:RepositoryUniqueid>1.3.6.1.4...1000</ihe:RepositoryUniqueid>
            <ihe:DocumentUniqueid>1.3.6.1.4...2300</ihe:DocumentUniqueid>
            <Rows>300</Rows>
            <Columns>300</Columns>
            <ContentTypeList>
              <ContentType>image/jpeg</ContentType>
            </ContentTypeList>
          </ihe:DocumentRequest>
          <ihe:DocumentRequest>
            <ihe:RepositoryUniqueid>1.3.6.1.4...1000</ihe:RepositoryUniqueid>
            <ihe:DocumentUniqueid>1.3.6.1.4...2301</ihe:DocumentUniqueid>
            <Rows>300</Rows>
            <Columns>300</Columns>
            <ContentTypeList>
              <ContentType>image/jpeg</ContentType>
            </ContentTypeList>
          </ihe:DocumentRequest>
        </SeriesRequest>
      </StudyRequest>
    </RetrieveRenderedImagingDocumentSetRequest>
  </s:Body>
</s:Envelope>
```

```
--MIMEBoundaryurn_uuid_DCD262C64C22DB97351256303951323--
```

## E.3 WADO WS Response Example

Example of the response corresponding to the above request:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: multipart/related; boundary=MIMEBoundaryurn_uuid_F862C3E04D9E35266C1256303956115;
  type="application/xop+xml"; start="0.urn:uuid:F862C3E04D9E35266C1256303956116@apache.org";
  start-info="application/soap+xml"; action="urn:ihe:iti:2007:RetrieveDocumentSetResponse"
Date: Fri, 23 Oct 2009 13:19:11 GMT
```

--MIMEBoundaryurn\_uuid\_F862C3E04D9E35266C1256303956115  
 Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"  
 Content-Transfer-Encoding: binary  
 Content-ID: <0.urn:uuid:F862C3E04D9E35266C1256303956116@apache.org>

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      urn:ihe:iti:2007:RetrieveRenderedImagingDocumentSetResponse</a:Action>
    <a:RelatesTo>urn:uuid:0fbfdced-6c01-4d09-a110-2201afedaa02</a:RelatesTo>
  </s:Header>
  <s:Body>
    <RetrieveDocumentSetResponse
      xmlns="urn:ihe:iti:xds-b:2007"
      xmlns:lcm="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:3.0"
      xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
      xmlns:rsm="urn:oasis:names:tc:ebxml-regrep:xsd:rsm:3.0"
      xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0">
      <rs:RegistryResponse status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success"/>
      <DocumentResponse>
        <ihe:RepositoryUniqueid>1.3.6.1.4...1000</ihe:RepositoryUniqueid>
        <SourceDocumentUniqueid>1.3.6.1.4...2300</SourceDocumentUniqueid>
        <Annotation>patient</Annotation>
        <Rows>300</Rows>
        <Columns>300</Columns>
        <Region>
          <Xmin>0.0</Xmin>
          <Ymin>0.0</Ymin>
          <Xmax>1.0</Xmax>
          <Ymax>1.0</Ymax>
        </Region>
        <WindowCenter>2000</WindowCenter>
        <WindowWidth>4096</WindowWidth>
        <ImageQuality>30</ImageQuality>
        <mimeType>image/jpeg</mimeType>
        <Document>
<xop:Include href="cid:1.urn:uuid:F862C3E04D9E35266C1256303956117@apache.org"
  xmlns:xop="http://www.w3.org/2004/08/xop/include"/>
        </Document>
      </DocumentResponse>
    </RetrieveDocumentSetResponse>
  </s:Body>
</s:Envelope>
<xop:Include href="cid:2.urn:uuid:F862C3E04D9E35266C1256303956117@apache.org"
  xmlns:xop="http://www.w3.org/2004/08/xop/include"/>
```

```
</Document>
</DocumentResponse>
</RetrieveDocumentSetResponse>
</s:Body>
</s:Envelope>
```

```
--MIMEBoundaryurn_uuid_F862C3E04D9E35266C1256303956115
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <1.urn:uuid:F862C3E04D9E35266C1256303956117@apache.org>
```

This is the binary JPEG payload for the first image.

```
--MIMEBoundaryurn_uuid_F862C3E04D9E35266C1256303956115
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <2.urn:uuid:F862C3E04D9E35266C1256303956117@apache.org>
```

This is the binary JPEG payload for the second image.

```
--MIMEBoundaryurn_uuid_F862C3E04D9E35266C1256303956115—
```



# F DICOM JSON Model

## F.1 Introduction to JavaScript Object Notation (JSON)

JSON is a text-based open standard, derived from JavaScript, for representing data structures and associated arrays. It is language-independent, and primarily used for serializing and transmitting lightweight structured data over a network connection. It is described in detail by the Internet Engineering Task Force (IETF) in RFC 4627, available at <http://www.ietf.org/rfc/rfc4627.txt>.

The DICOM JSON Model complements the XML-based Native DICOM Model, by providing a lightweight representation of data returned by DICOM web services. While this representation can be used to encode any type of DICOM Data Set it is expected to be used by client applications, especially mobile clients, such as described in the QIDO-RS use cases (see Chapter HHH in PS3.17).

## F.2 DICOM JSON Model

The DICOM JSON Model follows the Native DICOM Model for XML very closely, so that systems can take advantage of both formats without much retooling. The Media Type for DICOM JSON is application/json. The default character repertoire shall be UTF-8 / ISO\_IR 192.

### F.2.1 Multiple Results Structure

Multiple results returned in JSON are organized as a single top-level array of JSON objects. This differs from the Native DICOM Model, which returns multiple results as a multi-part collection of singular XML documents.

#### F.2.1.1 Examples

##### F.2.1.1.1 Native DICOM Model

```
<?xml version="1.0" encoding="UTF-8" xml:space="preserve" ?>
<NativeDicomModel>
  <DicomAttribute tag="0020000D" vr="UI" keyword="StudyInstanceUID">
    <Value number="1">1.2.392.200036.9116.2.2.2.1762893313.1029997326.945873</Value>
  </DicomAttribute>
</NativeDicomModel>
...
<?xml version="1.0" encoding="UTF-8" xml:space="preserve" ?>
<NativeDicomModel>
  <DicomAttribute tag="0020000D" vr="UI" keyword="StudyInstanceUID">
    <Value number="1">1.2.444.200036.9116.2.2.2.1762893313.1029997326.945876</Value>
  </DicomAttribute>
</NativeDicomModel>
```

##### F.2.1.1.2 DICOM JSON Model

```
[
  {
    "0020000D": {
      "vr": "UI",
      "Value": [ "1.2.392.200036.9116.2.2.2.1762893313.1029997326.945873" ]
    }
  },
  {
    "0020000D": {
      "vr": "UI",
      "Value": [ "1.2.392.200036.9116.2.2.2.2162893313.1029997326.945876" ]
    }
  }
]
```

```
}  
]
```

## F.2.2 DICOM JSON Model Object Structure

The DICOM JSON Model object is a representation of a DICOM Data Set.

The internal structure of the DICOM JSON Model object is a sequence of objects representing attributes within the DICOM Data Set.

Attribute objects within a DICOM JSON Model object must be ordered by their property name in ascending order.

Group Length (gggg,0000) attributes shall not be included in a DICOM JSON Model object.

The name of each attribute object is:

- The eight character uppercase hexadecimal representation of a DICOM Tag

Each attribute object contains the following named child objects:

- **vr**: A string encoding the DICOM Value Representation. The mapping between DICOM Value Representations and JSON Value Representations is described in Section F.2.3.

- At most one of:

- **Value**: An array containing one of:

- The Value Field elements of a DICOM attribute with a VR other than PN, SQ, OB, OD, OF, OW, or UN (described in Section F.2.4)

The encoding of empty Value Field elements is described in Section F.2.5

- The Value Field elements of a DICOM attribute with a VR of PN. The non-empty name components of each element are encoded as a JSON strings with the following names:

- Alphabetic
- Ideographic
- Phonetic

- JSON DICOM Model objects corresponding to the sequence items of an attribute with a VR of SQ

Empty sequence items are represented by empty objects

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

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

Note

1. For Private Data Elements, the group and element numbers will follow the rules specified in Section 7.8.1 in PS3.5
2. The person name representation is more closely aligned with the DICOM Data Element representation than the DICOM PS3.19 XML representation.

## F.2.3 DICOM JSON Value Representation

The value representation (VR) is included in each DICOM JSON Model attribute object and named "vr". For example:

"vr": "CS"

All DICOM Value Representations are mapped to specified JSON Data Types (see Table F.2.3-1). The JSON encodings shall conform to the Definition, Character Repertoire (if applicable) and Length of Value specified for that Value Representation (see Section 6.2 "Value Representation (VR)" in PS3.5) with the following exceptions:

- Attributes with a Value Representation of AT shall be restricted to eight character uppercase hexadecimal representation of a DICOM Tag

**Table F.2.3-1. DICOM VR to JSON Data Type Mapping**

VR Name	Type	JSON Data Type
AE	Application Entity	<i>String</i>
AS	Age String	<i>String</i>
AT	Attribute Tag	<i>String</i>
CS	Code String	<i>String</i>
DA	Date	<i>String</i>
DS	Decimal	<i>Number</i>
DT	Date Time	<i>String</i>
FL	Floating Point Single	<i>Number</i>
FD	Floating Point Double	<i>Number</i>
IS	Integer String	<i>Number</i>
LO	Long String	<i>String</i>
LT	Long Text	<i>String</i>
OB	Other Byte String	<i>Base64 encoded string</i>
OD	Other Double String	<i>Base64 encoded string</i>
OF	Other Float String	<i>Base64 encoded string</i>
OW	Other Word String	<i>Base64 encoded string</i>
PN	Person Name	<i>Object containing Person Name component groups as strings (see Section F.2.2)</i>
SH	Short String	<i>String</i>
SL	Signed Long	<i>Number</i>
SQ	Sequence	<i>Array containing DICOM JSON Objects</i>
SS	Signed Short	<i>Number</i>
ST	Short Text	<i>String</i>
TM	Time	<i>String</i>
UI	UID	<i>String</i>
UL	Unsigned Long	<i>Number</i>
UN	Unknown	<i>Base64 encoded string</i>
UR	URI	<i>String</i>
US	Unsigned Short	<i>Number</i>
UT	Unlimited Text	<i>String</i>

Although data, such as dates, are represented in the DICOM JSON model as strings, it is expected that they will be treated in the same manner as the original attribute as defined by Chapter 6 in PS3.6.

## F.2.4 DICOM JSON Value Multiplicity

The value or values of a given DICOM attribute are given in the "Value" array. The value multiplicity (VM) is not contained in the DICOM JSON object.

For example:

```
"Value": [ "bar", "foo" ]
```

or:

```
"Value": [ "bar" ]
```

## F.2.5 DICOM JSON Model Null Values

If an attribute is present in DICOM but empty (i.e., Value Length is 0), it shall be preserved in the DICOM JSON attribute object containing no "Value", "BulkDataURI" or "InlineBinary".

If a multi-valued attribute has one or more empty values these are represented as "null" array elements. For example:

```
"Value": [ "bar", null, "foo" ]
```

If a sequence contains empty items these are represented as empty JSON object in the array.

```
"Value": [ { ... }, { }, { ... } ]
```

## F.2.6 BulkDataURI

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

## F.2.7 InlineBinary

If an attribute contains an "InlineBinary", this contains the base64 encoding of the enclosing attribute's Value Field.

There is a single InlineBinary value representing the entire Value Field, and not one per Value in the case where the Value Multiplicity is greater than one. 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 with a VL of 8192 and a VM of 4096 would both be represented as a single InlineBinary string.

All rules (e.g., byte ordering and swapping) in DICOM PS3.5 apply.

### Note

Implementers should in particular pay attention to the PS3.5 rules regarding the value representations of OD, OF and OW.

## F.3 Transformation with other DICOM Formats

### F.3.1 Native DICOM Model XML

The transformation between the Native DICOM Model XML and the DICOM JSON model cannot be done through the use of generic XML - JSON converters.

The mapping between the two formats is as follows (see also Table F.3.1-1):

- The XML "NativeDicomModel" element maps to the DICOM JSON Model Object
- Each "DicomAttribute" element maps to an attribute object within the DICOM JSON model object
  - The "tag" attribute maps to the JSON object name
    - The Native DICOM Model XML allows for duplicate Tag values and the DICOM JSON model does not. To resolve this, private attribute Tag values must be remapped according to the conflict avoidance rules specified in Section 7.8.1 "Private Data Element Tags" in PS3.5.

- The "vr" attribute maps to the "vr" child string
- "Value" elements map to members of the "Value" child array
  - A "Value" element with the attribute "number=n" maps to "Value[n-1]"
  - Empty "Value" elements are represented by "null" entries in the "Value" array
- "PersonName" elements map to objects within the "Value" array. For a "PersonName" element with the attribute "number=n":
  - The "Alphabetic" element maps to "Value[n-1].Alphabetic"
  - The "Ideographic" element maps to "PersonName[n].Ideographic"
  - The "Phonetic" element maps to "PersonName[n].Phonetic"
- "Item" elements map to members of the "Value" child array
  - An "Item" element with the attribute "number=n" maps to "Value[n-1]"
  - Empty "Item" elements are represented by empty JSON property entries in the "Value" array
- The "uri" attribute of the "BulkData" element maps to the "BulkDataURI" string
- The "InlineBinary" element maps to the "InlineBinary" string

**Table F.3.1-1. XML to JSON Mapping**

DICOM PS3.19 XML	DICOM JSON Model
<pre> &lt;NativeDicomModel&gt; &lt;DicomAttribute tag=  ggggee01  ... /&gt; &lt;DicomAttribute tag=  ggggee02  ... /&gt; ... &lt;/NativeDicomModel&gt; </pre>	<pre> {   ggggee01  : { ... },   ggggee02  : { ... },   ... } </pre>
<pre> &lt;DicomAttribute tag=  ggggeeee vr=  VR  &gt; &lt;Value number="1"&gt;  Value  &lt;/Value&gt; &lt;/DicomAttribute&gt; </pre>	<pre> ggggeeee  : {   "vr":  VR  ,   "Value": [  Value  ] } </pre>
<pre> &lt;DicomAttribute tag=  ggggeeee  ... &gt; &lt;Value number="1"&gt;  Value1  &lt;/Value&gt; &lt;Value number="2"&gt;  Value2  &lt;/Value&gt; ... &lt;/DicomAttribute&gt; </pre>	<pre> ggggeeee  : {   ...   "Value": [  Value1  ,     Value2  , ...   ] } </pre>

DICOM PS3.19 XML	DICOM JSON Model
<code>&lt;DicomAttribute tag= ggggeeee ... &gt;</code> <code>&lt;/DicomAttribute&gt;</code>	<code>ggggeeee : {</code> <code>...</code> <code>}</code>

DICOM PS3.19 XML	DICOM JSON Model
<pre> &lt;DicomAttribute tag= <b>gggggeeee</b> vr="PN" ... &gt;   &lt;PersonName number="1"&gt;     &lt;Alphabetic&gt;       &lt;FamilyName&gt; <b>SB1</b>     &lt;/FamilyName&gt;     &lt;GivenName&gt; <b>SB2</b>     &lt;/GivenName&gt;     &lt;MiddleName&gt; <b>SB3</b>     &lt;/MiddleName&gt;     &lt;NamePrefix&gt; <b>SB4</b>     &lt;/NamePrefix&gt;     &lt;NameSuffix&gt; <b>SB5</b>     &lt;/NameSuffix&gt;     &lt;/Alphabetic&gt;     &lt;Ideographic&gt;       &lt;FamilyName&gt; <b>ID1</b>     &lt;/FamilyName&gt;     ...     &lt;/Ideographic&gt;     &lt;Phonetic&gt;       &lt;FamilyName&gt; <b>PH1</b>     &lt;/FamilyName&gt;     ...     &lt;/Phonetic&gt;   &lt;/PersonName&gt;   &lt;PersonName number="2"&gt;     &lt;Alphabetic&gt;       &lt;FamilyName&gt; <b>SB6</b>     &lt;/FamilyName&gt;     &lt;/Alphabetic&gt;   &lt;/PersonName&gt; &lt;/DicomAttribute&gt; </pre>	<pre> <b>gggggeeee</b> : {   ...   "vr": "PN",   "Value": [     {       "Alphabetic" : "<b>SB1^SB2^SB3^SB4^SB5</b>",       "Ideographic": "<b>ID1^ID2^ID3^ID4^ID5</b>" ,       "Phonetic": "<b>PH1^PH2^PH3^PH4^PH5</b>"     },     {       "Alphabetic":         "<b>SB6</b>"     }   ] } </pre>

DICOM PS3.19 XML	DICOM JSON Model
<pre> &lt;DicomAttribute tag= ggggeeee vr="SQ" ... &gt; &lt;Item number="1"&gt; &lt;DicomAttribute tag= ggggee01 ... /&gt; &lt;DicomAttribute tag= ggggee02 ... /&gt; ... &lt;/Item&gt; &lt;Item number="2"&gt; &lt;DicomAttribute tag= ggggee01 ... /&gt; &lt;DicomAttribute tag= ggggee02 ... /&gt; ... &lt;/Item&gt; &lt;Item number="3"&gt; &lt;/Item&gt; ... &lt;/DicomAttribute&gt; </pre>	<pre> ggggeeee : { ... "vr": "SQ", "Value": [ { ggggeee01 : { ... }, ggggeee02 : { ... }, ... } { ggggeee01 : { ... }, ggggeee02 : { ... }, ... } { } ... ] } </pre>
<pre> &lt;DicomAttribute tag= ggggeeee ... &gt; &lt;BulkData URI= BulkDataURI &gt; &lt;/DicomAttribute&gt; </pre>	<pre> ggggeeee : { ... "BulkDataURI": BulkDataURI } </pre>
<pre> &lt;DicomAttribute tag= ggggeeee ... &gt; &lt;InlineBinary&gt; Base64String &lt;/InlineBinary&gt; &lt;/DicomAttribute&gt; </pre>	<pre> ggggeeee : { ... "InlineBinary": " Base64String" } </pre>
<pre> &lt;DicomAttribute tag= gggg00ee PrivateCreator= PrivateCreator ... &gt; ... &lt;/DicomAttribute&gt; </pre>	<pre> ggggXXee : { ... } </pre>

## F.4 DICOM JSON Model Example

// The following example is a QIDO-RS SearchForStudies response consisting

// of two matching studies, corresponding to the example QIDO-RS request:  
 // GET <http://qido.nema.org/studies?PatientID=12345&includefield=all&limit=2>

```
[
  { // Result 1
    "00080005": {
      "vr": "CS",
      "Value": [ "ISO_IR192" ]
    },
    "00080020": {
      "vr": "DT",
      "Value": [ "20130409" ]
    },
    "00080030": {
      "vr": "TM",
      "Value": [ "131600.0000" ]
    },
    "00080050": {
      "vr": "SH",
      "Value": [ "11235813" ]
    },
    "00080056": {
      "vr": "CS",
      "Value": [ "ONLINE" ]
    },
    "00080061": {
      "vr": "CS",
      "Value": [
        "CT",
        "PET"
      ]
    },
    "00080090": {
      "vr": "PN",
      "Value": [
        {
          "Alphabetic": "^Bob^^Dr."
        }
      ]
    },
    "00081190": {
      "vr": "UR",
      "Value": [ "http://wado.nema.org/studies/1.2.392.200036.9116.2.2.2.1762893313.1029997326.945873" ]
    },
    "00090010": {
      "vr": "LO",
      "Value": [ "Vendor A" ]
    },
    "00091002": {
      "vr": "UN",
      "Value": [ "z0x9c8v7" ]
    },
    "00100010": {
      "vr": "PN",
      "Value": [
        {
          "Alphabetic": "Wang^XiaoDong",
          "Ideographic": "王^小東"
        }
      ]
    }
  ]
```

```

    },
    "00100020": {
      "vr": "LO",
      "Value": [ "12345" ]
    },
    "00100021": {
      "vr": "LO",
      "Value": [ "Hospital A" ]
    },
    "00100030": {
      "vr": "DT",
      "Value": [ "19670701" ]
    },
    "00100040": {
      "vr": "CS",
      "Value": [ "M" ]
    },
    "00101002": {
      "vr": "SQ",
      "Value": [
        {
          "00100020": {
            "vr": "LO",
            "Value": [ "54321" ]
          },
          "00100021": {
            "vr": "LO",
            "Value": [ "Hospital B" ]
          }
        },
        {
          "00100020": {
            "vr": "LO",
            "Value": [ "24680" ]
          },
          "00100021": {
            "vr": "LO",
            "Value": [ "Hospital C" ]
          }
        }
      ]
    },
    "0020000D": {
      "vr": "UI",
      "Value": [ "1.2.392.200036.9116.2.2.2.1762893313.1029997326.945873" ]
    },
    "00200010": {
      "vr": "SH",
      "Value": [ "11235813" ]
    },
    "00201206": {
      "vr": "IS",
      "Value": [ 4 ]
    },
    "00201208": {
      "vr": "IS",
      "Value": [ 942 ]
    }
  },
  { // Result 2

```

```

"00080005": {
  "vr": "CS",
  "Value": [ "ISO_IR192" ]
},
"00080020": {
  "vr": "DT",
  "Value": [ "20130309" ]
},
"00080030": {
  "vr": "TM",
  "Value": [ "111900.0000" ]
},
"00080050": {
  "vr": "SH",
  "Value": [ "11235821" ]
},
"00080056": {
  "vr": "CS",
  "Value": [ "ONLINE" ]
},
"00080061": {
  "vr": "CS",
  "Value": [
    "CT",
    "PET"
  ]
},
"00080090": {
  "vr": "PN",
  "Value": [
    {
      "Alphabetic": "^Bob^Dr."
    }
  ]
},
"00081190": {
  "vr": "UR",
  "Value": [ "http://wado.nema.org/studies/
1.2.392.200036.9116.2.2.2.2162893313.1029997326.945876" ]
},
"00090010": {
  "vr": "LO",
  "Value": [ "Vendor A" ]
},
"00091002": {
  "vr": "UN",
  "Value": [ "z0x9c8v7" ]
},
"00100010": {
  "vr": "PN",
  "Value": [
    {
      "Alphabetic": "Wang^XiaoDong",
      "Ideographic": "王^小東"
    }
  ]
},
"00100020": {
  "vr": "LO",
  "Value": [ "12345" ]
}

```

```

    },
    "00100021": {
      "vr": "LO",
      "Value": [ "Hospital A" ]
    },
    "00100030": {
      "vr": "DT",
      "Value": [ "19670701" ]
    },
    "00100040": {
      "vr": "CS",
      "Value": [ "M" ]
    },
    "00101002": {
      "vr": "SQ",
      "Value": [
        {
          "00100020": {
            "vr": "LO",
            "Value": [ "54321" ]
          },
          "00100021": {
            "vr": "LO",
            "Value": [ "Hospital B" ]
          }
        },
        {
          "00100020": {
            "vr": "LO",
            "Value": [ "24680" ]
          },
          "00100021": {
            "vr": "LO",
            "Value": [ "Hospital C" ]
          }
        }
      ]
    },
    "0020000D": {
      "vr": "UI",
      "Value": [ "1.2.392.200036.9116.2.2.2.2162893313.1029997326.945876" ]
    },
    "00200010": {
      "vr": "SH",
      "Value": [ "11235821" ]
    },
    "00201206": {
      "vr": "IS",
      "Value": [ 5 ]
    },
    "00201208": {
      "vr": "IS",
      "Value": [ 1123 ]
    }
  }
]

```

## F.5 References

IETF RFC 4627 <http://www.ietf.org/rfc/rfc4627.txt> (Normative JSON definition)

JSON. <http://www.json.org/> (Informative)

Wikipedia, definition of JSON. <http://en.wikipedia.org/wiki/JSON> (Informative)

JSON in FHIR. <http://www.hl7.org/implement/standards/fhir/formats.htm#json> (Informative)



# G WADL JSON Representation

## G.1 Introduction

While the WADL specification only specifies an XML encoding for the WADL payload, the data structure can easily be represented using JSON. Additionally, conversion from XML to JSON and vice-versa can be done in a lossless manner.

## G.2 XML Elements

The JSON encoding of WADL XML elements depends on whether the element is:

- a "doc" element
- an element that is unique within a particular parent element (e.g., "request")
- an element that can be repeated within a particular parent element (e.g., "param")

### G.2.1 Doc Elements

A "doc" element is represented as an array of objects, where each object may contain:

- a "@xml:lang" string
- a "@title" string
- a "value" string

Example:

```
"doc": [  
  {  
    "@xml:lang": "en",  
    "value": "Granular cell tumor"  
  },  
  {  
    "@xml:lang": "ja",  
    "value": "顆粒細胞腫"  
  },  
  {  
    "@xml:lang": "fr",  
    "value": "Tumeur à cellules granuleuses"  
  }  
]
```

### G.2.2 Unique Elements

All unique WADL XML elements are represented as an object whose name is the name of the XML element and where each member may contain:

- a "@{attribute}" string for each XML attribute of the name {attribute}
- a child object for each child element that must be unique
- a child array for each child element that may not be unique

Example:

```
"request": {  
  "param": [ ... ],  
  "representation": [ ... ]  
}
```

### G.2.3 Repeatable Elements

All repeatable WADL XML elements are represented as an array of objects whose name is the name of the XML element and where each may contain:

- a "@{attribute}" string for each XML attribute of the name {attribute}
- a child object for each child element that must be unique
- a child array for each child element that may not be unique

Example:

```
"param": [  
  {  
    "@name": "Accept",  
    "@style": "header"  
  },  
  {  
    "@name": "Cache-control",  
    "@style": "header"  
  }  
]
```