Digital Imaging and Communications in Medicine (DICOM)

Supplement 163: STore Over the Web by REpresentations State Transfer (REST) Services (STOW-RS)

Prepared by:

DICOM Standards Committee, Working Group 27 Web Technology

1300 N. 17th Street, Suite 1752

Rosslyn, Virginia 22209 USA

Contact: svastagh@medicalimaging.org

VERSION: Final Text, 19 August 2013

Developed in accordance with: DICOM Workitem 2011-09-B

Table of Contents

Scope and Field of Application

This Supplement defines Representational State Transfer (REST) Services for storing DICOM objects to DICOM SCPs, such as PACS, Vendor Neutral Archives (VNA), or clinical viewers.

While DICOM supports Web based access to studies using the WADO-URI, WADO-WS and WADO-RS standards, it does not currently support a Web based method of adding or updating studies. This supplement defines DICOM RESTful Services for storing PS 3.10 binary instances, as well as metadata plus bulk data.

RESTful change management mechanisms are not defined within this supplement.

Security is beyond the scope of the services defined in this supplement. However there are accepted industry guidelines on the use of generic Web security mechanisms with protected healthcare information (see DICOM PS 3.15).

Changes to NEMA Standards Publication PS 3.1-2011

Digital Imaging and Communications in Medicine (DICOM)

Part 1: Introduction and Overview

Insert into PS 3.1 Section 4 Symbols and Abbreviations

REST Representational State Transfer

RESTful A RESTful Web service is a Web service implemented using REST architecture and HTTP (see thesis at http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf).

STOW-RS STore Over the Web by RESTful Services

WADO-URI Web Access to DICOM Objects by URI

WADO-WS Web Access to DICOM Objects by Web Services (WS*)

WADO-RS Web Access to DICOM Objects by RESTful Services

Update PS 3.1 Section 6.1 as indicated

PS 3.18: ~~Web Access to DICOM Persistent Objects (WADO)~~ Web Services

Update PS 3.1 Section 6.18 as indicated

6.18 PS 3.18: WEB SERVICES

PS 3.18 of the DICOM Standard specifies the means whereby **Web Services** ~~**an HTTP request that includes a reference in the form of a URI/URL**~~ **can be used** for ~~**accessing**~~ **retrieving or storing to** ~~**a**~~ DICOM ~~**persistent**~~ object**s.** ~~**can be expressed as an HTTP URI/URL, WS or RESTful Web Service (RS) request that includes a pointerreference to a specific DICOM persistent object in the form of its Instance UID in the form of a URI/URL that denotes the object(s) being retrieved or stored.**~~

Requests that retrieve data specify the media type (format) of the response body. Requests that store data specify the media type of the request body.

The ~~**parameters of the query URL HTTP requests**~~ as defined within this standard are sufficient for the HTTP server to act as a DICOM SCU (Service Class User) to retrieve **or store** the requested object**s** from an appropriate DICOM SCP (Service Class Provider) using baseline DICOM functionality as defined in PS 3.4 and PS 3.7**, which is to say that the HTTP server can act as a proxy for the DICOM SCP.**

Changes to NEMA Standards Publication PS 3.2-2011

Digital Imaging and Communications in Medicine (DICOM)

Part 2: Conformance

Insert into PS 3.2 Section 4 Symbols and Abbreviations (in alphabetical order)

REST **Representational State Transfer**

RESTful **A RESTful Web service is a Web service implemented using REST architecture and HTTP (see thesis at http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf).**

STOW-RS **STore Over the Web by RESTful Services**

WADO-URI **Web Access to DICOM Objects by URI**

WADO-WS **Web Access to DICOM Objects by Web Services (WS*)**

WADO-RS **Web Access to DICOM Objects by RESTful Services**

Update to PS 3.2 Appendix A.4.x.y "Application Entity <1>

A.4.2.x.y **~~WADO-RS~~ RESTful Services Specifications**

All **~~WADO~~** RESTful services that are supported shall be listed. Other **~~WADO~~** RESTful services that are not supported may be indicated.

For each supported service, any parameters and restrictions on those parameters shall be described.

Any connection policies such as restrictions on the number of connections, support for pipeline requests, etc. shall be described.

Append after PS 3.2 Annex I (Informative) CONFORMANCE STATEMENT SAMPLE WADO SERVICE

ANNEX J (Informative) CONFORMANCE STATEMENT SAMPLE STOW SERVICE

Disclaimer:

This document is an example DICOM Conformance Statement for a fictional application service called EXAMPLE-STOW-SERVICE produced by a fictional vendor called EXAMPLE-PACS-PRODUCTS.

As stated in the annex title, this document is truly informative, and not normative. A conformance statement of an actual product might implement additional services and options as appropriate for its specific purpose. In addition, an actual product might implement the services described in a different manner and, for example, with different characteristics and/or sequencing of activities. In other words, this conformance statement example does not intend to standardize a particular manner that a product might implement DICOM functionality.

J.0 COVER PAGE

Company Name: EXAMPLE-PACS-PRODUCTS-VENDOR

Product Name: EXAMPLE-STOW-SERVICE

Version: 1.0-rev. A.1

Internal document number: 1024-1960-xx-yy-zz rev 1

Date: YYYYMMDD

J.1 CONFORMANCE STATEMENT OVERVIEW

This fictional product EXAMPLE-STOW-SERVICE implements the STOW-RS services for storing DICOM SOP Instances into an EXAMPLE-PACS-ARCHIVE. The EXAMPLE-STOW-SERVICE is only available as a plug in option for the EXAMPLE-PACS-ARCHIVE. All of the networking, database, and other services are provided by the EXAMPLE-PACS-ARCHIVE. This conformance claim refers to the conformance claim for the EXAMPLE-PACS-ARCHIVE for all such services.

Table J.1-1 provides an overview of the network services supported by EXAMPLE-STOW-SERVICE.

Table J.1-1
NETWORK SERVICES

Network Service	User of Service (Client)	Provider of Service (Server)
STorage Over the Web (STOW)		
STOW-RS – Store Instances	No	Yes

J.2 TABLE OF CONTENTS

A table of contents shall be provided to assist readers in easily finding the needed information.

J.3 INTRODUCTION

J.3.1 REVISION HISTORY

Document Version	Date of Issue	Author	Description
1.1	November 16th, 2012	LCS	Version for Final Text
1.2	February 8th, 2013	LCS	Revised Introduction
1.3	May 16th, 2013	SAR	Incorporated CP-71

J.3.2 AUDIENCE, REMARKS, TERMS AND DEFINITIONS, BASICS OF DICOM COMMUNICATION, ABBREVIATIONS, REFERENCES

See example text in Annex A.3.

J.3.3 ADDITIONAL REMARKS FOR THIS EXAMPLE

This document is a sample DICOM Conformance Statement created for DICOM PS 3.2. It is to be used solely as an example to illustrate how to create a DICOM Conformance Statement for a DICOM Service Class Provider (SCP). The subject of the document, EXAMPLE-STOW-SERVICE, is a fictional product.

J.4 NETWORKING

J.4.1 IMPLEMENTATION MODEL

J.4.1.1 Application Data Flow

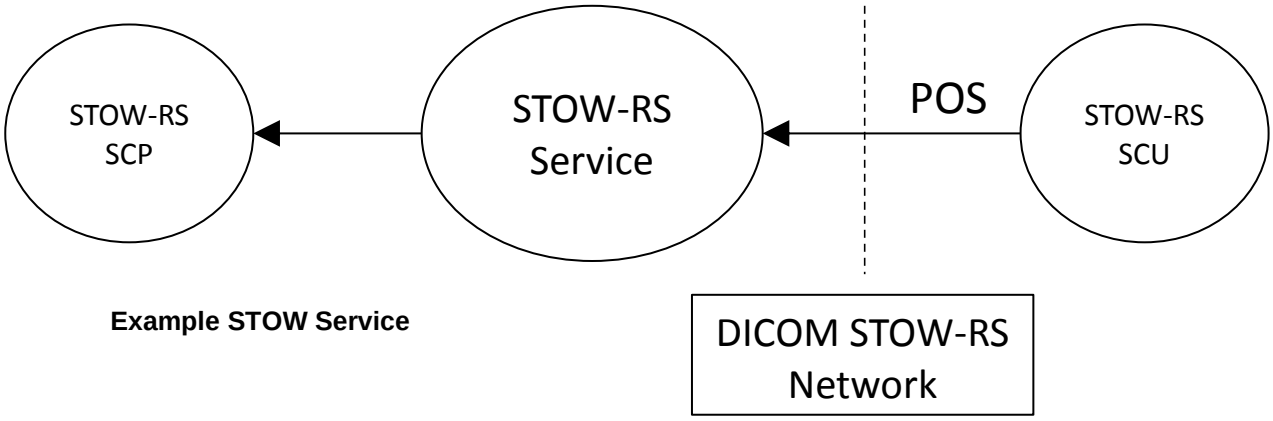


Figure J.4.1-1
APPLICATION DATA FLOW DIAGRAM

The STOW-RS Service Application receives STOW requests from a remote AE. These requests are HTTP/1.1 POST requests. It is associated with the local real-world activity “Store Instances”. It converts these requests into internal functions to store the given SOP Instances. It returns a summary HTTP status line, including a status code and an associated textual phase, followed by an XML message indicating success, warning, or failure for each instance to the requesting remote AE.

J.4.1.2 Functional Definition of AEs

J.4.1.2.1 Functional Definition of STOW Service Application

The reception of a STOW-RS POST request will activate the STOW-RS Service. The storage request is based upon the accept headers in the STOW-RS POST request. The response includes an HTTP/1.1 status line, including a status-code and its associated textual phrase, followed by an XML message indicating success, warning, or failure for each instance stored by the STOW-RS service.

J.4.2 AE SPECIFICATIONS

This AE complies with PS 3.18, Section 6, specification for STOW-RS storage.

J.4.2.1 STOW-RS Specifications

J.4.2.1.1 STOW-RS Store Instance

Table J.4.2-1
STOW-RS STORE INSTANCES SPECIFICATION

Category	Restrictions
Media Types Supported (Accept header)	Restricted to application/dicom or application/dicom+xml
Transfer Syntaxes Supported (Media Type parameter)	Any Transfer Syntax supported by the hosting EXAMPLE-PACS-ARCHIVE
SOP Class Restrictions	Restricted to SOP classes supported by the hosting EXAMPLE-PACS-ARCHIVE
Size restriction	Restricted to size supported by the hosting EXAMPLE-PACS-ARCHIVE

J.4.2.2.4 Connection Policies

J.4.2.2.4.1 General

All standard RS connection policies apply. There are no extensions for RS options.

J.4.2.2.4.2 Number of Connections

EXAMPLE-STOW-SERVICE limits the number of simultaneous RS requests. Additional requests will be queued after the HTTP/1.1 connection is accepted. When an earlier request completes, a pending request will proceed.

Table J.4.2-4
NUMBER OF HTTP REQUESTS SUPPORTED

Maximum number of simultaneous RS requests	100 (configurable)

J.4.2.2.4.3 Asynchronous Nature

EXAMPLE-STOW-SERVICE does not support RS asynchronous response.

J.4.2.2.4.4 SOP Specific Conformance for SOP Class(es)

The EXAMPLE-STOW-SERVICE response message header contains status codes indicating success, warning, or failure as shown in the "HTTP/1.1 Standard Response Codes" below. No additional status codes are used.

HTTP/1.1 STANDARD RESPONSE CODES

Service Status	HTTP/1.1 Status Code	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 authenticated.
	403 – Forbidden	This indicates that the STOW-RS Service understood the request, but is refusing to fulfill it (e.g. an authenticated user with insufficient privileges).
	409 – Conflict	This indicates that the STOW-RS Service request was formed correctly but the service was unable to store any instances due to a conflict in the request (e.g. unsupported SOP Class or StudyInstanceUID mismatch). This may also be used to indicate that a STOW-RS Service was unable to store any instances for a mixture of reasons. Additional information regarding the instance errors can be found in the XML response message body.
	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.

The EXAMPLE-STOW-SERVICE response message body (PS 3.18 XML Store Instances Response Module) contains the DICOM status codes for individual SOP Instances indicating success, warning, or failure as defined below. No additional status codes are used.

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

B000 – Coercion of Data Elements

The STOW-RS Service modified one or more data elements during storage of the instance.

B006 – Elements Discarded
The STOW-RS Service discarded some data elements during storage of the instance.

B007 – Data Set does not match SOP Class
The STOW-RS Service stored the instance despite the Data Set not matching the constraints of the SOP Class.

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 is used.

For the following semantics the associated value are used for the Failure Reason (0008,1197).

A700 – Refused out of Resources
The STOW-RS Service did not store the instance because it was out of memory.

A710 – Refused out of Resources
The STOW-RS Service did not store the instance because it was out of storage space.

A900 – Error: Data Set does not match SOP Class
The STOW-RS Service did not store the instance because the SOP Class of an element in the Referenced SOP Instance Sequence did not correspond to the SOP class registered for this SOP Instance at the STOW-RS Service.

C000 – Error: Cannot understand
The STOW-RS Service did not store the instance because it cannot understand certain Data Elements.

C122 – Referenced Transfer Syntax not supported
The STOW-RS Service did not store the instance because it does not support the requested Transfer Syntax for the instance.

0110 – Processing failure
The STOW-RS Service did not store the instance because of a general failure in processing the operation.

0122 – Referenced SOP Class not supported
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 errors.

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

J.4.3 NETWORK INTERFACES

J.4.3.1 Physical Network Interface

EXAMPLE-STOW-SERVICE uses the network interface from the hosting EXAMPLE-PACS-ARCHIVE. See its conformance claim for details.

J.4.3.2 Additional Protocols

EXAMPLE-STOW-SERVICE uses the network services from the hosting EXAMPLE-PACS-ARCHIVE. See its conformance claim for details.

J.4.3.3 IPv4 and IPv6 Support

This product supports both IPv4 and IPv6 connections.

J.4.4 CONFIGURATION

J.4.4.1 STOW-RS Interface

The EXAMPLE-STOW-SERVICE is configured to respond to TLS protected traffic. The TLS port will refuse any connection from a system that is not recognized as authenticated by a known authority.

J.5 Media Interchange

Not applicable

J.6 SUPPORT OF CHARACTER SETS

All EXAMPLE-STOW-SERVICEs support Unicode UTF-8 for all RS transactions.

The EXAMPLE-STOW-SERVICE does not convert character sets when storing PS 3.10 binary Instances. The original DICOM encoded character sets are preserved.

J.7 SECURITY

The EXAMPLE-STOW-SERVICE supports the following transport level security measures:

- HTTP BASIC Authorization over SSL
- Digest Authorization
- SSL Client Certificates

The transport level security measures support bi-directional authentication using TLS connections. The EXAMPLE-STOW-SERVICE can provide its certificate information, and can be configured with either a direct comparison (self-signed) certificate or a chain of trust certificates.

The EXAMPLE-STOW-SERVICE will refuse a connection over TLS from a source that does not have a recognized authentication. For example, a certificate authenticated by "Big Hospital Provider" will not be accepted unless the EXAMPLE-STOW-SERVICE has been configured to accept authentications from "Big Hospital Provider". The list of acceptable certificates for EXAMPLE-STOW-SERVICE is not shared with certificates used by other system applications and must be maintained independently.

The EXAMPLE-STOW-SERVICE can optionally be configured to use the following session authentication mechanisms:

- Kerberos Local Domain Sessions
- Shibboleth Cross Domain Sessions (using SAML2.0)
- OAuth 2.0 complying with IHE ITI Internet User Authentication (IUA)

J.8 ANNEXES

J.8.1 IOD CONTENTS

See conformance claim for the EXAMPLE-PACS-ARCHIVE.

J.8.2 DATA DICTIONARY OF PRIVATE ATTRIBUTES

No data dictionary for private attributes is provided. Private attributes are stored as received without modification.

J.8.3 CODED TERMINOLOGY AND TEMPLATES

See conformance claim for EXAMPLE-PACS-ARCHIVE.

J.8.4 STANDARD EXTENDED / SPECIALIZED / PRIVATE SOP CLASSES

See conformance claim for EXAMPLE-PACS-ARCHIVE.

J.8.5 PRIVATE TRANSFER SYNTAXES

Private transfer syntaxes are not supported.

Changes to NEMA Standards Publication PS 3.6-2011

Digital Imaging and Communications in Medicine (DICOM)

Part 6: Data Dictionary

Item: Add the following Data Element to Part 6 Section 6:

Registry of DICOM data elements

Tag	Name	Keyword	VR	VM
(0008,1190)	Retrieve URL	RetrieveURL	UT	1
(0008,1196)	Warning Reason	WarningReason	US	1

Changes to NEMA Standards Publication PS 3.17-2011

Digital Imaging and Communications in Medicine (DICOM)

Part 17: Explanatory Information

Update PS 3.17 Annex HHH as indicated.

Annex HHH - Evolution of WADO to Web and REST Services (Informative)

This annex discusses the design considerations that went into the definition of the WADO extension to Web **and REST** services.

HHH.1 REQUEST AND RESPONSE PARAMETERS

HHH.1.1 Request Parameters

The new service based on WS should continue to support all the request parameters defined by WADO, for maintaining backward compatibility with the present URI based WADO, including the options to return either native DICOM objects or a rendered object (JPEG, PDF etc.). ~~**These are summarized as below:**~~

The WADO-RS **and STOW-RS** request**s** ha~~**s**~~**ve** no parameters because data is requested through well defined URLs and content negotiation through HTTP headers.

The WADO-WS request parameters are summarized as below:

Update PS 3.17 HHH.1.2 Response parameters as indicated.

HHH.1.2 Response parameters

HHH.1.2.1 ~~URI~~ WADO-URI

In the URI-based WADO, the response is the single payload returned in the HTTP Get response. It may be the DICOM object in a DICOM format or in a rendered format.

HHH.1.2.2 WADO-WS

In the Web Services implementation, for the “DICOM Requester” and the “Rendered Requester” transactions, one or more DICOM objects are returned using the MTOM/XOP **(http://www.w3.org/TR/soap12-mtom)** mechanism as well as associated metadata.

For the “Metadata Requester” transaction, the response will contain the XML encoded part containing the information selected from the retrieved objects header using the “XPath” filter as described in the Native DICOM Model defined in PS 3.19.

HHH.1.2.3 WADO-RS

The WADO-RS Service is a transport service**, as opposed to a rendering service, which** provides resources **that** enable machine to machine transfers of**binary instances**, pixel data, bulk data, and metadata. **These services are not primarily intended to be directly displayable in a browser.**

In the REST Services implementation:

- For the "DICOM Requester", one or more multipart/related items are returned containing ~~the~~ ~~DICOM~~ **PS 3.10 binary** instances of a Study**,** ~~or~~ Series, or a**n individual DICOM SOP** ~~single~~ Instance.
- For the "Frame Pixel Data Requester", one or more multipart/related items are returned containing the pixel data of a multi-frame SOP Instance.
- For the "Bulk Data Requester", one or more multipart/related items are returned containing the bulk data of a Study, Series or SOP Instance.
- For the "Metadata Requester", an item is returned containing the XML encoded metadata selected from the retrieved objects header as described in the Native DICOM Model defined in PS 3.19.

HHH.1.2.4 STOW-RS

The STOW-RS Service provides the ability to STore Over the Web using RESTful Services (i.e. ~~HTTP~~**HTTP/1.1 based functionality equivalent to C-Store).**

- **For the "DICOM Creator", one or more multipart/related parts** ~~items~~ **are stored (posted to a STOW-RS Service) containing one or more DICOM Composite SOP Instances.**
- **For the "Metadata and Bulk Data Creator", one or more multipart/related parts are stored (posted to a STOW-RS Service) containing the XML encoded metadata defined in PS 3.19 and one or more parts containing the bulk data of a Study, Series or SOP Instance.**

Update PS 3.17 HHH.2 Web Services Implementation as indicated

HHH.2 WEB AND REST SERVICES IMPLEMENTATION

The implementation architecture has to maximize interoperability, preserve or improve performance and minimize storage overhead.

The Web and REST Services technologies have been selected to:

a. be firewall friendly and supporting security,

b. be supported by and interoperable between multiple development environments, and

c. have sufficient performance for both large and small text and for binary data.

The XML implementation of the messages uses the CamelCase parameter style used in SOAP 1.2 (element names starting with an upper case character, e.g., ElementOne, attribute names starting with a lower case character e.g. attributeOne).

The WADO-WS response will be provided as a list of instances in MTOM/XOP ("DICOM" or "Rendered" Requesters), XML encoded additional information resulting from the XPath filters applied on every objects selected ("Information Requester")

The WADO-RS response will be provided as a list of XML and/or binary ~~objects~~ **instances** in a multipart/related response. The type of response depends on the **media types listed in the** Accept ~~type in the HTTP H~~**h**eader.

The STOW-RS response is a standard HTTP/1.1 ~~response code~~**status line and possibly an XML response message body. The meaning of the success, warning, or failure statuses are defined in PS 3.18, Section 6.**

Update PS 3.17 HHH.3 Uses for Web Services as indicated

HHH.3 USES FOR WADO-WS ~~WEB~~ AND WADO-RS SERVICES

HHH.3.1 General requirements

Imaging information is important in the context of EMR/EHR. But EMR/EHR systems often do not support the DICOM protocol. The EMR/EHR vendors need access using web and web service technologies to satisfy their users.

HHH.3.2 Analysis of use cases

Examples of use cases / clinical scenarios, as the basis to develop the requirements, include:

1. Providing access to images and reports from a point-of-service application e.g., EMR.
2. Following references to significant images used to create an imaging report and displaying those images.
3. Following references / links to relevant images and imaging reports in email correspondence or clinical reports e.g., clinical summary.
4. Providing access to anonymized DICOM images and reports for clinical research and teaching purposes.
5. Providing access to a DICOM encoded imaging report associated with the DICOM IE (patient/study/series/objects) to support remote diagnostic workflows e.g., urgent medical incidents, remote consultation, clinical training, teleradiology/telemedicine applications.
6. Providing access to summary or selected information from DICOM objects.
7. Providing access to complete studies for caching, viewing, or image processing.
8. **Storing DICOM SOP Instances using HTTP over a Network from PACS to PACS, from PACS to VNA, from VNA to VNA, from clinical application to PACS, or any other DICOM SCP.**
9. **Web clients, including mobile ones, retrieving XML and bulk data from a WADO-RS Service and adding new instances to a study.**

Examples of the use cases described in 1 above are:

a. The EMR displays in JPEG one image with annotations on it (patient and/or technique related), based upon information provided in a report.
b. The EMR retrieves from a “Manifest” document all the referenced objects in DICOM and launches a DICOM viewer for displaying them (use case addressed by the IHE XDS-I.b profile).
c. The EMR displays in JPEG one image per series with information describing every series (e.g. series description).
d. The EMR displays in JPEG all the images of a series with information describing the series as well as every image (e.g. instance number and slice location for scanner images).
e. The EMR populates in its database for all the instances referred in a manifest (KOS) the relevant information (study ID/UID/AccessionNumber/Description/DateTime, series UID/Modality/Description/DateTime, instance UID/InstanceNumber/SliceLocation).
f. The EMR displays patient demographics and image slices in a browser by accessing studies through URLs that are cached and rendered in a remote data center.
g. **A hospital transfers a DICOM Study over a network to another healthcare provider without needing special ports opened in either firewall.**
h. **A diagnostic visualization client, during post-processing, adds a series of Instances containing measurements, annotations, or reports.**
i. **A healthcare provider transfers a DICOM Study to a Patient Health Record (PHR) at the request of the patient.**

Append below HHH3.3.8 Metadata Requestor

HHH.3.3.9 DICOM Creator

A. The requesting system is an application capable of making HTTP/1.1 Service requests and able to process data encoded as PS 3.10 binary instances.

B. The request specifies

1. The STOW-RS Service to store POST requests.
2. Optionally, it may also specify Study Instance UID indicating all POST requests are for the indicated study.
3. SOP Instances, per DICOM PS 3.10 encoding.

C. The response is a standard HTTP/1.1 status line and an XML response message body. The meaning of the success, warning, or failure statuses are defined in PS 3.18, Section 6.

HHH.3.3.10 Metadata and Bulk Data Creator

A. The requesting system is an application capable of making HTTP/1.1 requests and able to process data encoded as PS 3.19 XML metadata.

B. The request specifies

1. The STOW-RS Service to store POST requests.
2. Optionally, it may also specify Study Instance UID indicating all POST requests are for the indicated study.
3. XML metadata, per DICOM PS 3.19 encodings, and bulk data.

C. The response is a standard HTTP/1.1 status line and an XML response message body. The meaning of the success, warning, or failure statuses are defined in PS 3.18, Section 6.

Changes to NEMA Standards Publication PS 3.18-2011

Digital Imaging and Communications in Medicine (DICOM)

Part 18: Web Services

Append to PS 3.18 Section 3 Normative references

3 Normative references

IETF RFC822 Standard for ARPA Internet Text Messages

Insert into PS 3.18 Section 5 Symbols and abbreviated terms (in correct alphabetical order)

5 Symbols and abbreviated terms

RESTful **A RESTful Web service is a Web service implemented using REST architecture and HTTP (see thesis at http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf).**

WADO-URI **Web Access to DICOM Objects by URI**

WADO-WS **Web Access to DICOM Objects by Web Services (WS*)**

WADO-RS **Web Access to DICOM Objects by RESTful Services**

STOW-RS **STore Over the Web by RESTful Services**

Update PS 3.18 Section 6.1 INTERACTION as indicated.

6 Data Communication Requirements

6.1 INTERACTION

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

Multiple communications modes are possible:

— URI based ~~**mechanism**~~ using HTTP Get: WADO**-URI** ~~**Type**~~ 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. Pixel Data Requester (Retrieve Instance Frame Pixel Data)

c. Bulk Data Requester (Retrieve Study, Series, Instance bulk data)

d. Metadata Requester (Retrieve Study Metadata)

— **RESTful Services (RS) using HTTP POST: STOW-RS, either:**

a. DICOM Creator (Store Instances)

b. Metadata and Bulk Data Creator (Store Instances)

Update PS 3.18 Section 6.3 as indicated

6.3 WADO-URI RESPONSE

Update PS 3.18 Section 6.4 as indicated

6.4 WADO-WS REQUEST/RESPONSE

Update PS 3.18 Section 6.5 as indicated

6.5 WADO-RS REQUEST/RESPONSE

Update PS 3.18 Section 6.5.1 as indicated

6.5.1 WADO- RS – RetrieveStudy

Update PS 3.18 Section 6.5.2 as indicated

6.5.2 WADO-RS – RetrieveSeries

Update PS 3.18 Section 6.5.3 as indicated

6.5.3 WADO-RS – RetrieveInstance

Update PS 3.18 Section 6.5.4 as indicated

6.5.4 WADO-RS – RetrieveFrames

Update PS 3.18 Section 6.5.5 as indicated

6.5.5 WADO-RS – RetrieveBulkdata

Update PS 3.18 Section 6.5.6 as indicated

6.5.6 WADO-RS – RetrieveMetadata

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

The full study metadata includes all attributes of the study; 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.

Notes: 1. The server is not required to replace any attribute with a BulkDataURL; this is intended to allow the server to provide clients with metatdata 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.

Append to PS 3.18 Section 6.5 RS REQUEST/RESPONSE

6.6 STOW-RS REQUEST/RESPONSE

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

1. Store Instances

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

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

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

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

- All XML request messages shall be encoded as described in the Native DICOM Model defined in PS 3.19 with one message part per XML object.
- 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 RFC2616 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 PS 3.10 binary instances.
        - multipart/related; type=application/dicom+xml; boundary={messageBoundary}

          Specifies that the post is PS 3.19 XML metadata and bulk data.

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 PS 3.5, Section 9 and Appendix B.

644 **6.6.1.1.1 DICOM Request Message Body**

The DICOM Request Message has a multipart body.

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

6.6.1.1.2 Metadata and Bulk Data Request Message Body

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

- Content-Type:
    - 656 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 658 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.
- 660 Each body part is either DICOM PS 3.19 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.
- 662 The first part in the multipart request will contain the following HTTP headers:
    - Content-Type: application/dicom+xml; transfer-syntax={TransferSyntaxUID}
- 664 Subsequent items will contain the following HTTP headers (order is not guaranteed):
    - additional metadata with the following headers:
        - 666 Content-Type: application/dicom+xml; transfer-syntax={TransferSyntaxUID}
    - an uncompressed bulk data element encoded in Little Endian binary format with the 668 following headers:
        - Content-Type: application/octet-stream
        - 670 Content-Location: {BulkDataURI}
    - a compressed pixel data object from a SOP Instance in the Study with the following 672 headers:
        - Content-Type: {MediaType}
        - 674 Content-Location: {BulkDataURI}
- Metadata and its associated bulk data shall always be sent in the same POST request.

676 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.

678

6.6.1.2 Action

680 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,
682 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
684 occurs.

If any element is coerced or corrected, the Original Attribute Sequence (0400,0561) shall be included in
686 the DICOM Object that is stored and may be included in the PS 3.18 XMLStore Instances Response Module in the response.

688 Note: For more information on populating the Original Attribute Sequence, see PS 3.3, Section C.12.1 SOP Common.

690

6.6.1.3 Response

692 The RESTful Service shall return an HTTP status line, including a status code and associated textual phrase for the entire set of stored SOP Instances, followed by an XML message body containing a DICOM
694 PS 3.19 XML representation of the Store Instances Response Module as defined in Table 6.6.1-2.

6.6.1.3.1 Response Status Line

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

698 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,
700 the response code shall be a 409 and will be followed by the PS 3.18 XML Store Instances Response Module as the message body, which contains additional information regarding instance errors .

702 In all other conditions, the RESTful Service shall return an "HTTP 202 – Accepted" response code. Additional information regarding the instance warnings or failures will be found in the PS 3.18 XML Store
704 Instances Response Module in the response message body.

Table 6.6.1-1
706 **HTTP/1.1 STANDARD RESPONSE CODE**

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

		mixture of reasons. Additional information regarding the instance errors can be found in the XML response message body.
	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 6.6.1.3.3. For example,

```
Warning: "A700: Out of memory"
```

6.6.1.3.2 Response Message Body

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

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

Table 6.6.1-2 defines the Attributes for referencing SOP Instances which are contained in a PS 3.18 XML 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.
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.
>Include 'SOP Instance Reference Macro' Table 10-11			
>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.
>Include 'SOP Instance Reference Macro' Table 10-11			
>Retrieve URL	(0008,1190)	2	The URL where the SOP Instance is available for retrieval via a WADO-URI or WADO-RS service. If the study Retrieve URI is specified above, this URI can be constructed if the client knows the series and instance UIDs.
>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 which 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.			

722

6.6.1.3.2.1 Store Instances Response Attribute Description

724 #### 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):

726 B000 – Coercion of Data Elements

The STOW-RS Service modified one or more data elements during storage of the
728 instance. See Section 6.6.1.3.

B006 – Elements Discarded
The STOW-RS Service discarded some data elements during storage of the instance. See Section 6.6.1.3.

B007 – Data Set does not match SOP Class
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:

A7xx – Refused out of Resources
The STOW-RS Service did not store the instance because it was out of resources.

A9xx – Error: Data Set does not match SOP Class
The STOW-RS Service did not store the instance because the instance does not conform to its specified SOP Class. Cxxx – Error: Cannot understand
The STOW-RS Service did not store the instance because it cannot understand certain Data Elements.

C122 – Referenced Transfer Syntax not supported
The STOW-RS Service did not store the instance because it does not support the requested Transfer Syntax for the instance.

0110 – Processing failure
The STOW-RS Service did not store the instance because of a general failure in processing the operation.

0122 – Referenced SOP Class not supported
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 PS 3.18 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="UT" keyword="RetrieveURL">
            <Value
    number="1">https://wadors.hospital.com/studies/2.16.124.113543.6003.1154777499.30246.19789.3503
    430045/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="UT" keyword="RetrieveURL">
            <Value
    number="1">https://wadors.hospital.com/studies/2.16.124.113543.6003.1154777499.30246.19789.3503
    430045/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="UT" keyword="RetrieveURL">
        <Value
    number="1">https://wadors.hospital.com/studies/2.16.124.113543.6003.1154777499.30246.19789.3503
    430045</Value>
      </DicomAttribute>

    </NativeDicomModel>
```

Changes to NEMA Standards Publication PS 3.19-2011

Digital Imaging and Communications in Medicine (DICOM)

Part 19: Application Hosting

Update PS 3.19 Annex A.1.1

A.1.1 Usage

The Native DICOM Model defines a representation of binary-encoded DICOM ~~**SOP Instances**~~ **Data Sets** as XML Infosets that allows a recipient of data to navigate through a binary DICOM data set using XML-based tools instead of relying on toolkits that understand the binary encoding of DICOM.

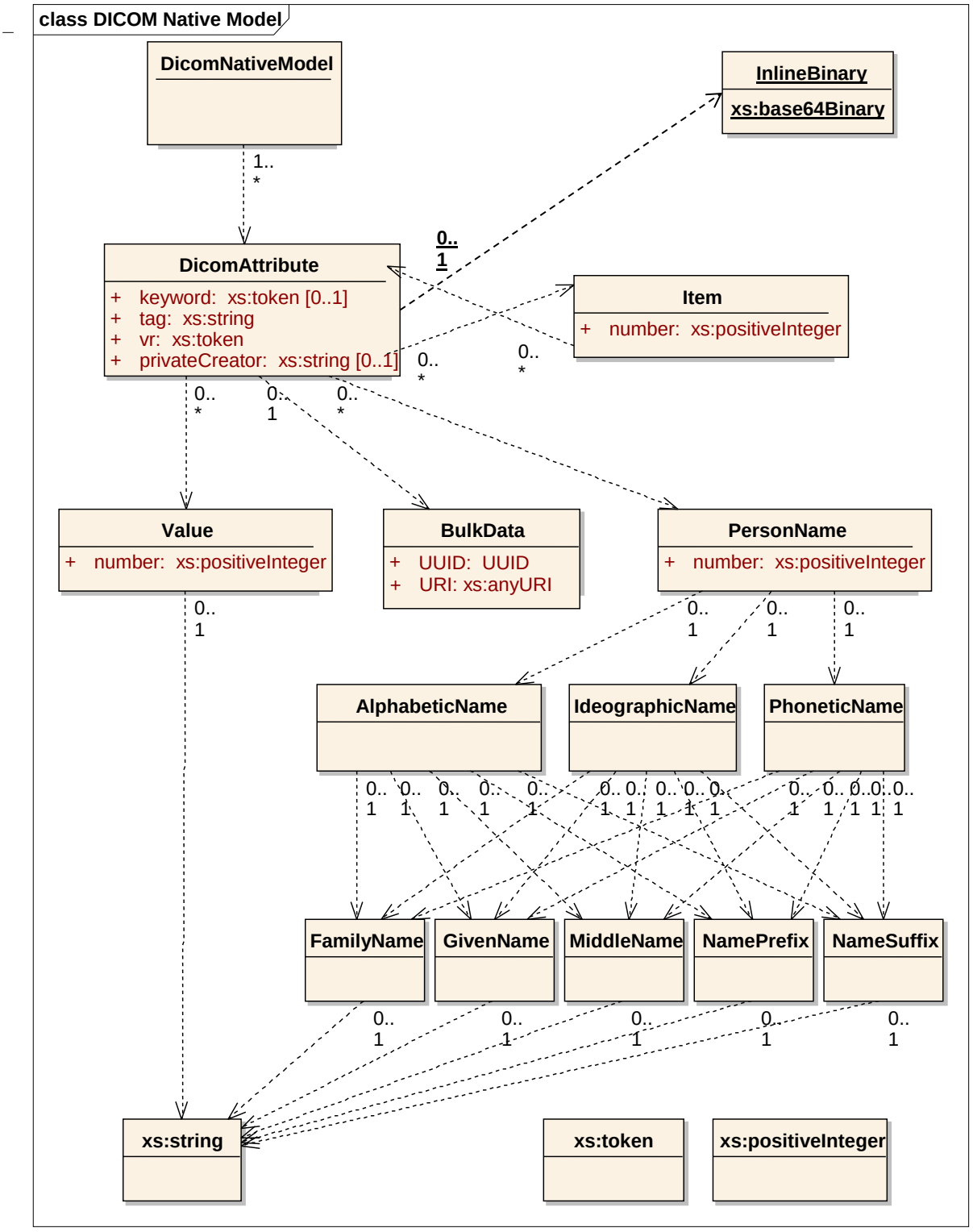
Note: It is not the intention that this form be utilized as the basis for other uses. This form does not take advantage of the self-validation features that could be possible with a pure XML representation of the data.

With the exception of padding, a data source that is creating a new ~~**instance**~~ **data set** of a native DICOM Model (e.g. the result from some analysis application) shall follow the DICOM encoding rules (e.g. the handling of character sets) in creating Values for the DicomAttributes within the instance of the DICOM Native Model.

Update PS 3.19 A.1.4 Information as shown below.

A.1.4 Information Model

A diagram of the Native DICOM Model appears in Figure A.1.4-1.



Update PS 3.19 Annex A.1.5

A.1.5 Description

866 **Table A.1.5-1 Native DICOM Model**

Name	Optionality	Cardinality	Description
NativeDicomModel	R	1	An Infoset (as defined in W3C Recommendation XML Information Set "http://www.w3.org/TR/xml-infoset/") representing the content of a DICOM Data Set (as defined in PS 3.5)~~, which may be either:~~. **The directive `xml:space="preserve"` shall be included.** **Examples include:** - the contents of an entire DICOM Composite Instance (as defined in PS 3.3) in response to a native model request, or - the contents of part of a DICOM Composite Instance in response to a query on a native model, or **- the contents of a STOW-RS response** - the contents of a Sequence Item (as defined in PS 3.5), recursively included within an Infoset Value element. ~~**The directive `xml:space="preserve"` shall be included.**~~
Include 'DICOM DataSet Macro' Table A.1.5-2			

868 **Update Bulk Data URI in PS 3.19 Table A.1.5-2 DICOM Data Set Macro**

Table A.1.5-2 DICOM Data Set Macro

Name	Optionality	Cardinality	Description
DicomAttribute	O	0-n	An Infoset element corresponding to each DICOM Attribute.
>keyword	C	A	The keyword as defined in PS 3.6. Required unless the DICOM Data Element is unknown to the host.
…			
>Value	C	1-n	A Value from the Value Field of the DICOM Data Element. There is one Infoset Value element for each DICOM

			Value or Sequence Item. Required if the DICOM Data Element represented is not zero length and an Item, PersonName, **InlineBinary,** or BulkData XML element is not present. Shall not be used if the VR of the enclosing Attribute is either SQ or PN.
…			
>BulkData	C	1	A reference to a blob of data that the recipient may retrieve through use of the GetData() method**~~, or a~~** WADO-RS **call, or a STOW-RS call.** Required if the DICOM Data Element represented is not zero length and an XML Infoset Value, Item, **InlineBinary,** or PersonName element is not present. The provider of the data may use a BulkData reference at its discretion to avoid encoding a large DICOM Value Field by value in the Infoset**, f~~F~~or** example, **~~a provider may include large binary values such as~~** pixel data or look up tables**~~, which typically would be located in a file, as BulkData references~~**. **~~Note that t~~T**here is a single BulkData Infoset element representing the entire Value Field, and not one per Value in the case where the Value Multiplicity is greater than one. 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 BulkData element. All rules (e.g. byte ordering and swapping) in DICOM PS 3.5 apply. Note: Implementers should in particular pay attention the PS 3.5 rules regarding the value representations of OW and OF. If the BulkData has a string or text Value Representation, the value(s) of the DICOM Specific Character Set Data Element, if present, might be necessary to

			determine its encoding.
>>UUID	C	A	An identifier of this bulk data reference formatted as a UUID using the hexadecimal representation defined in ITU-T Recommendation X.667. Required if BulkData URI is not present. Shall not be present otherwise.

>>URI	C	A	The HTTP(S) URI for this bulk data reference. Required if the NativeDicomModel was: - returned in response to a WADO-RS Retrieve Metadata request **- uploaded as part of a STOW-RS request** Shall not be present otherwise.
>InlineBinary	**C**	**1**	**The Value Field of the enclosing Attribute encoded as base64.** **Required if the DICOM Data Element represented is:** • **not zero length** • **the VR if the enclosing Attribute is either OB, OD, OF, OW, or UN** • **an XML Infoset Value or BulkData XML element is not present** **Shall not be present otherwise.** **~~Note that~~ tThere is a single InlineBinary Infoset element representing the entire Value Field, and not one per Value in the case where the Value Multiplicity is greater than one. 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 element.** **All rules (e.g. byte ordering and swapping) in DICOM PS 3.5 apply.** **Note: Implementers should in particular pay attention the PS 3.5 rules regarding the value representations of OD, OF and OW.**

Update PS 3.19 A.1.6 Schema as shown below.

A.1.6 Schema

The Normative version of the XML Schema for the Native DICOM Model follows:

```
default namespace="http://dicom.nema.org/PS3.19/models/NativeDICOM"
```

```
# This schema was created as an intermediary, a means of describing
# native binary encoded DICOM objects as XML Infosets, thus allowing
# one to manipulate binary DICOM objects using familiar XML tools.
# As such, the schema is designed to facilitate a simple, mechanical,
# bi-directional translation between binary encoded DICOM and XML-like
# constructs without constraints, and to simplify identifying portions
# of a DICOM object using XPath statements.
#
# Since this schema has minimal type checking, it is neither intended
# to be used for any operation that involves hand coding, nor to
# describe a definitive, fully validating encoding of DICOM concepts
# into XML, as what one might use, for example, in a robust XML
# database system or in XML-based forms, though it may be used
# as a means for translating binary DICOM Objects into such a form
# (e.g. through an XSLT script).

start = element NativeDicomModel { DicomDataSet }

# A DICOM Data Set is as defined in PS3.5.  It does not appear
# as an XML Element, since it does not appear in the binary encoded
# DICOM objects.  It exists here merely as a documentation aid.
DicomDataSet = DicomAttribute*

DicomAttribute = element DicomAttribute {
  Tag, VR, Keyword?, PrivateCreator?,
  ( BulkData | Value+ | Item+ | PersonName+ | InlineBinary )?
}
BulkData = element BulkData{ (UUID | URI) }
Value = element Value { Number, xsd:string }
InlineBinary = element InlineBinary { xsd:base64Binary }
Item = element Item { Number, DicomDataSet }
PersonName = element PersonName {
  Number,
  element SingleByte  { NameComponents }?,
  element Ideographic { NameComponents }?,
  element Phonetic    { NameComponents }?
}

NameComponents =
  element FamilyName {xsd:string}?,
  element GivenName  {xsd:string}?,
  element MiddleName {xsd:string}?,
  element NamePrefix {xsd:string}?,
  element NameSuffix {xsd:string}?

# keyword is the attribute tag from PS3.6
# (derived from the DICOM Attribute's name)
Keyword = attribute keyword { xsd:token }
# canonical XML definition of Hex, with lowercase letters disallowed
Tag = attribute tag { xsd:string{ minLength="8" maxLength="8" pattern="[0-9A-F]{8}" } }
VR = attribute vr { "AE" | "AS" | "AT"| "CS" | "DA" | "DS" | "DT" | "FL" | "FD"
                  | "IS" | "LO" | "LT" | "OB" | "OF" | "OW" | "PN" | "SH" | "SL"
                  | "SQ" | "SS" | "ST" | "TM" | "UI" | "UL" | "UN" | "US" | "UT" }
PrivateCreator = attribute privateCreator{ xsd:string }
UUID = attribute uuid { xsd:string }
URI = attribute uri { xsd:anyURI }
Number = attribute number { xsd:positiveInteger }
```