

5

10

**Digital Imaging and Communications in Medicine (DICOM)**

*Supplement 194: RESTful Non-Patient Instance Storage*

15

20 **DICOM Standards Committee**

1300 N. 17th Street Suite 900

Rosslyn, Virginia 22209 USA

VERSION: Letter Ballot

Developed in accordance with work item 2015-12-C.

25

## Closed Issues

#	Description
1	Should the service allow retrieving multiple instances in the same transaction? <b>No</b> , for symmetry
2	Should the service allow storing multiple instances in the same transaction? <b>Yes</b> , for symmetry.
3	Do we want accept and charset query parameters? <b>Yes</b> , for symmetry with other services.
4	Do we need to support any transfer syntax other than EVRLE? <b>Yes</b> , for symmetry (deflate).
5	Is the distinction between Metadata and Bulkdata necessary? <b>Yes</b> , for symmetry and because some instances could be large (e.g. implants).
6	Should any other IODs be included? <b>No</b> .
7	Do we want/need a notification API for create, retrieve, update, events? Maybe for some, such as procedure protocols (need longer lived notifications), or could use search? <b>No</b> , not yet.
8	Should this service support only DICOM Media Types? <b>Yes</b> .
9	Do any of these IODs have Transfer Syntaxes that are specific to them? <b>No</b> , except 'deflate' and EVRLE which applies to all of them.
10	Is there any reason to Search inside individual Instances? <b>No</b> .
11	Should all DICOM Media Types ('application/dicom', 'application/dicom+xml', 'application/dicom+json') be supported? <b>Yes</b> .
12	Should we define the attributes that must be supported for search or leave them unspecified? Should each NPI IOD define its own Search attributes? What does CP 1550 do? <b>No</b> . Use the attributes defined in PS3.4.
13	Should support for the Store transaction be required? <b>No</b> , for symmetry with Studies service
14	The sections on status codes will be updated as part of Supplement 183: Web Services Re-Documentation
15	Should support for 'application/dicom' media type be required. <b>Yes</b> .
16	Should there be one /implant-templates or three different roots? <b>Decision</b> : One template.
17	Should the Information Model name be the root or should it have /npis root before the Information Model name. For example: /npis/color-palette, or /color-palette <b>Decision</b> : The format should be /{npi-name}

## Table of Contents

30	1	Scope and Field of Application	4
	6.X	RS Non-Patient Instance (NPI) Storage	4
	6.X.1	Resources	4
	6.X.2	General Query Parameters	5
	6.X.2.1	Accept	5
35	6.X.2.2	Character Set	5
	6.X.3	Transactions	5
	6.X.3.1	Retrieve Capabilities Transaction	5
	6.X.3.1.1	Request	5
	6.X.3.1.1.1	Resource	5

40	6.X.3.1.1.2	Query Parameters	6
	6.X.3.1.1.3	Request Header Fields	6
	6.X.3.1.1.4	Request Payload	6
	6.X.3.1.2	Behavior	6
	6.X.3.1.3	Response	6
45	6.X.3.1.3.1	Status Codes	6
	6.X.3.1.3.2	Response Header Fields	6
	6.X.3.1.3.3	Response Payload	6
	6.X.3.2	Retrieve Transaction	6
	6.X.3.2.1	Request	6
50	6.X.3.2.1.1	Resources	7
	6.X.3.2.1.2	Query Parameters	7
	6.X.3.2.1.3	Request Header Fields	7
	6.X.3.2.1.4	Request Payload	7
	6.X.3.2.2	Behavior	7
55	6.X.3.2.3	Response	7
	6.X.3.2.3.1	Status Codes	7
	6.X.3.2.3.2	Response Header Fields	8
	6.X.3.2.3.3	Response Payload	8
	6.X.3.3	Store Transaction	8
60	6.X.3.3.1	Request	8
	6.X.3.3.1.1	Resources	8
	6.X.3.3.1.2	Query Parameters	8
	6.X.3.3.1.3	Request Header Fields	8
	6.X.3.3.1.4	Request Payload	9
65	6.X.3.3.2	Behavior	9
	6.X.3.3.3	Response	9
	6.X.3.3.3.1	Status Codes	9
	6.X.3.3.3.2	Response Header Fields	10
	6.X.3.3.3.3	Response Payload	10
70	6.X.3.4	Search Transaction	10
	6.X.3.4.1	Request	10
	6.X.3.4.1.1	Resources	10
	6.X.3.4.1.2	Query Parameters	10
	6.X.3.4.1.3	Request Header Fields	10
75	6.X.3.4.1.4	Request Payload	11
	6.X.3.4.2	Behavior	11
	6.X.3.4.3	Response	11
	6.X.3.4.3.1	Status Codes	11
	6.X.3.4.3.2	Response Header Fields	11
80	6.X.3.4.3.3	Response Payload	11
	6.X.4	Media Types	11
	6.X.5	Conformance	12

# 1 Scope and Field of Application

This supplement defines Restful Services (RS) for retrieving, storing, and searching for non-patient related IODs such as hanging protocols, color palettes, procedure protocols, etc.

The transactions defined for this service are like those defined for the RS Studies Service. They allow a user agent to retrieve, store, and search for non-patient related IODs from an origin server in DICOM Media Types.

90 Security is beyond the scope of the RESTful services defined in this supplement. However, generic Web security mechanisms are fully compatible.

Images of phantoms, QC targets, doorknob swab photos, etc., are expected to be handled as pseudo-patients, because image IODs incorporate the patient/study hierarchy.

95 *Add the following section to Part 18, Section 6:*

## 6.X RS Non-Patient Instance (NPI) Storage

The RS Non-Patient Instances (NPI) Services define a set of RESTful transactions that enable a user agent to retrieve, store, and search an origin server for instances that are not specifically related to any patient.

100 An NPI Service manages a collection of resources belonging to the categories specified in 6.X.1. All NPI services shall support the Retrieve Capabilities, Retrieve, and Search transactions. Support for the Store transaction is optional.

### 6.X.1 Resources

An NPI Service manages resources from the same NPI Category. The target resource URIs have the following templates:

105     /{npi-name}  
       /{npi-name}/{uid}

Where

110     npi-name   = "color-palettes"  
                / "defined-procedure-protocols"  
                / "hanging-protocols"  
                / "implant-templates"  
    uid         ; is the Unique Identifier of an NPI Instance

Table 6.X.1-1 contains the templates for the NPI Resource Categories. It also includes the PS3.3 Section in which the corresponding IOD is defined.

115 **Table 6.X.1-1: Resource Categories, URI Templates and Descriptions**

Resource Category	URI Template and Description	IOD	Storage Class	Information Model
Color Palette	/color-palettes{/uid}	PS3.3, Annex A.58	PS3.4, Annex XX	PS3.4, Annex X.1.3
Defined Procedure Protocol	/defined-procedure-protocols{/uid}	PS3.3, Annex A.X1.2	PS.3.4, Annex XX	PS3.4, Annex BB.6.1
Hanging Protocol	/hanging-protocols{/uid}	PS3.3, Annex A.44	PS3.4, Annex XX	PS3.4, Annex U.1.3
Implant Template	/implant-templates{/uid}	PS3.3, Annex A.61	PS3.4, Annex XX	PS3.4, Annex BB.1.3

The NPI SOP Classes are listed in PS3.4, Table GG.3-1.

## 6.X.2 General Query Parameters

The Query Parameters in this section can be used with all NPI transactions.

### 6.X.2.1 Accept

120 The origin server shall support the Accept query parameter for all NPI transactions. See Section 6.1.1.5.

### 6.X.2.2 Character Set

The origin server shall support the Charset query parameter for all NPI transactions. See Section 6.1.2.2.

## 6.X.3 Transactions

The NPI Service defines the transactions listed in the following table:

125 **Table 6.X.3-1: NPI Service Transactions**

Transaction	Method	Resource	Payload		Description
			Request	Response	
Retrieve Capabilities	OPTIONS	/	N/A	Capabilities Description	Retrieves a description of the capabilities of the NPI Service, including transactions, resources, query parameters, etc.
Retrieve	GET	/ {npi-name} / {uid}	N/A	Instance and/or Status Report	Retrieves an Instance, specified by the target resource in an Acceptable DICOM Media Type.
Store	POST	/ {npi-name} / {uid}	Instance(s)	Status Report	Stores one or more DICOM Instances in a DICOM media type, contained in the request payload, in the location referenced by the target resource URL.
Search	GET	/ {npi-name} ? {params*}	N/A	Result(s) and/or Status Report	Searches the target resource for Instances that match the search parameters and returns a list of matches in an Acceptable DICOM Media Type.

Table 6.X.3-2 shows the target resources permitted for each transaction.

**Table 6.X.3-2 Resources by Transaction**

Resource	URI	Retrieve	Store	Search	Capabilities
NPI Service	/				X
All Instances	/ {npi-name}		X	X	
Instance	/ {npi-name} / {uid}	X	X		

### 6.X.3.1 Retrieve Capabilities Transaction

130 The Retrieve Capabilities transaction retrieves a machine-readable description of the NPI service implemented by an origin server. The response contains a machine-readable Capabilities Description. The Capabilities Description describes the transactions, resources, representations, etc. that are supported by the service(s).

An implementation of any NPI Service shall support the Retrieve Capabilities transaction.

#### 6.X.3.1.1 Request

135 The Retrieve Capabilities request uses the OPTIONS method and has the following format:

```

OPTIONS SP / SP version CRLF
Accept: 1#media-type CRLF
*(header-field CRLF)
CRLF
    
```

#### 140 6.X.3.1.1.1 Resource

The target resource for this transaction is the Base URI ("/") of the service.

### 6.X.3.1.1.2 Query Parameters

There are no additional Query Parameters.

### 6.X.3.1.1.3 Request Header Fields

145 Table 6.X.3-3 shows the most common Mandatory, Conditional, and Optional header fields for this transaction.

**Table 6.X.3-3: Request Header Fields**

Header Fields	Value	Usage	Description
Accept	media-range	M	See Section 6.1.1.7.
Accept-Charset	1#charset	O	See Section 6.1.2.3.

### 6.X.3.1.1.4 Request Payload

The request has no payload.

### 6.X.3.1.2 Behavior

150 The origin server shall return a machine-readable description of its capabilities in an Acceptable Media Type.

### 6.X.3.1.3 Response

The format of the response is as follows:

```

155   version SP status-code SP reason-phrase CRLF
      Content-Type: media-type CRLF
      *(header-field CRLF)
      CRLF
      payload
  
```

### 6.X.3.1.3.1 Status Codes

A success response shall have a status code of 200 (OK).

160 A failure response shall have a 400 or 500 level status code.

### 6.X.3.1.3.2 Response Header Fields

**Table 6.X.3-4: Response Header Fields**

Header Field	Value	Usage	Requirements
Content-Type	media-type	M	
Content-Length	uint	C	Shall be present if no transfer coding has been applied. Shall be absent otherwise.
Transfer-Encoding	encoding	C	Shall be present if a transfer coding has been applied. Shall be absent otherwise.
ETag	entity-tag	C	If the response status code is 200, the entity tag associated with the representation contained in the payload

### 6.X.3.1.3.3 Response Payload

165 A success response shall have a payload containing a Capabilities Description in the Selected Media Type. The Capabilities Description shall describe the service in as much detail as possible.

A failure response shall have a payload describing the error.

## 6.X.3.2 Retrieve Transaction

The Retrieve DICOM transaction retrieves the target NPI resource in a DICOM Media Type.

### 6.X.3.2.1 Request

170 The Retrieve request has the following syntax:

```
GET SP /{npi-name}/{uid} SP version CRLF
```

```

Accept: 1#dicom-media-type CRLF
[If-None-Match: entity-tag CRLF]
*(header-field CRLF)
CRLF
    
```

175

**6.X.3.2.1.1 Resources**

Table 6.X.3-5 shows the resources and URI Templates supported by the Retrieve DICOM transaction.

An origin server shall specify all supported resources in its conformance statement and in its response to the Retrieve Capabilities transaction.

180

**Table 6.X.3-5: Resources and URI Templates**

Resource	URI Template
Instance	/ {npi-name} / {uid}

**6.X.3.2.1.2 Query Parameters**

There are no additional query parameters.

**6.X.3.2.1.3 Request Header Fields**

Table 6.X.3-6 shows the most common Mandatory, Conditional, and common Optional header fields for this transaction.

185

**Table 6.X.3-6: URI Request Header Fields**

Header Field	Value	Usage
Accept	dicom-media-type	M

**6.X.3.2.1.4 Request Payload**

The request shall have no payload.

**6.X.3.2.2 Behavior**

190

The origin server shall locate the target resource and if found return it in an Acceptable DICOM Media Type.

**6.X.3.2.3 Response**

The response has the following syntax:

```

version SP status-code SP reason-phrase CRLF
Content-Type: dicom-media-type CRLF
[ETag: entity-tag CRLF]
[Last-Modified: HTTP-date CRLF]
*(header-field CRLF)
CRLF
Payload
    
```

195

**6.X.3.2.3.1 Status Codes**

200

The response shall have an appropriate status code. Table 6.X.3-7 contains the most common status codes for this transaction.

**Table 6.X.3-7: Status Codes**

Code	Description
200 (OK)	Indicates that the instance was successfully retrieved.
304 (Not Modified)	Indicates that the user agent's current representation is up to date, so no payload was returned. This status code shall only be returned for a Conditional Retrieve request containing an If-None-Match header field.
400 (Bad Request)	Indicates that the origin server did not store any of the representations contained in the request payload because of errors in the request message. For example, an invalid Query Parameter or an invalid SOP instance.

404 (Not Found)	Indicates that the origin server did not find a current representation for the target resource or is not willing to disclose that one exists. For example, an unsupported IOD, or SOP Instance not on server.
406 (Unsupported Media Type)	Indicates that the origin server does not support any of the Acceptable Media Types.

See [RFC7231, Section 6<<https://tools.ietf.org/html/rfc7231#section-6>>].

### 205 6.X.3.2.3.2 Response Header Fields

Table 6.X.3-8: Request Header Fields

Header Field	Value	Usage	Requirements
Content-Type	dicom-media-type	M	
Content-Length	uint	C	Shall be present if no transfer coding has been applied. Shall be absent otherwise.
Transfer-Encoding	encoding	C	Shall be present if a transfer coding has been applied. Shall be absent otherwise.
ETag	entity-tag	C	If the response status code is 200 or 204, this header field contains the entity-tag associated with the representations contained in the payload

### 6.X.3.2.3.3 Response Payload

A success response shall have a payload containing the DICOM instances specified by the target resource. The payload may be single part or multipart depending on the media type.

210 If there are any errors or warnings encountered by the origin server, the response shall contain a Status Report describing them.

## 6.X.3.3 Store Transaction

This transaction requests that the origin server store the representations of the NPIs contained in the request payload so that they may be retrieved in the future using the Instance UIDs.

215 The Store transaction is optional.

### 6.X.3.3.1 Request

Transactions in this service use the POST method. The request syntax is:

```

220 POST SP /{npi-name} {/uid} SP version CRLF
Content-Type: dicom-media-type CRLF
*(header-field CRLF)
CRLF
payload

```

### 6.X.3.3.1.1 Resources

225 The target URL shall reference either the All Instances or Instance resource. Table 6.X.3-9 shows the resources and URI Templates supported by the Store DICOM transaction.

Table 6.X.3-9: Resources and URI Templates

Resource	URI Template	Description
All Instances	/{npi-name}	Stores representations of a set of Instances.
Instance	/{npi-name} / {uid}	Stores a representation of a single Instance with a UID equal to uid.

### 6.X.3.3.1.2 Query Parameters

There are no additional Query Parameters.

### 6.X.3.3.1.3 Request Header Fields

230 Table 6.X.3-10: Store Request Header Fields

Header Field	Value	Usage	Requirements
--------------	-------	-------	--------------

Content-Type	dicom-media-type	M	
Accept	dicom-media-type	M	
Content-Length	uint	C	Shall be present if no transfer coding has been applied. Shall be absent otherwise.
Transfer-Encoding	encoding	C	Shall be present if a transfer coding has been applied. Shall be absent otherwise.

#### 6.X.3.3.1.4 Request Payload

The request payload shall be present and shall contain one or more representations in the DICOM Media Type specified by the Content-Type header field.

#### 6.X.3.3.2 Behavior

235 The origin server stores the representations contained in the request payload so that they may be retrieved later using the Retrieve DICOM transaction.

Before storing the representations, the origin server may coerce the values of data elements.

If any element is coerced, the Original Attribute Sequence (0400,0561) (see PS3.3, Section C.12.1) shall be included in the stored DICOM instances, and the response shall describe the modifications.

#### 240 6.X.3.3.3 Response

The response shall have the following syntax:

```

version SP status-code SP reason-phrase CRLF
*(header-field CRLF)
CRLF
245 [Status Report]
```

The response shall contain an appropriate status code and may contain a Status Report.

#### 6.X.3.3.3.1 Status Codes

The response shall have an appropriate status code. Table 6.X.3-11 contains the most common status codes for this transaction.

250

**Table 6.X.3-11: Common Status Codes**

Status Code	Description
200 (OK)	Indicates that the origin server successfully stored or created at least one of the representations contained in the request payload and is returning a response payload.
201 (Created)	Indicates that the origin server has successfully created at least one of the representations contained in the request payload and may be returning a response payload.
202 (Accepted)	Indicates that the origin server successfully validated the request message, but has not yet stored or created the representations in the request payload. The origin server may or may not have validated the payload. The user agent can use a Query or Retrieve transaction later to determine if the request has completed. Alternatively, if the Service supports notifications, and the user agent has a subscription for newly created resources, it will receive an event notification when the transaction has completed.
204 (No Content)	Indicates that the origin server has successfully stored all the representations contained in the request payload without any modifications and is not returning a response payload.
400 (Bad Request)	Indicates that the origin server did not store any of the representations contained in the request payload because of errors in the request message. For example, an invalid Query Parameter or an invalid SOP instance.
404 (Not Found)	Indicates that the origin server did not find a current representation for the target resource or is not willing to disclose that one exists. For example, an unsupported IOD, or SOP Instance not on server.

409 (Conflict)	Indicates that the request could not be completed due to a conflict with the current state of the target resource.
415 (Unsupported Media Type)	Indicates that the origin server does not support the media type specified in the Content-Type header field of the request, and none of the representations contained in the request were processed or stored.

### 6.X.3.3.3.2 Response Header Fields

**Table 6.X.3-12: Store Response Header Fields**

Header Field	Value	Usage	Requirements
Content-Type	dicom-media-type	M	
Content-Length	uint	C	Shall be present if no transfer coding has been applied. Shall be absent otherwise.
Transfer-Encoding	encoding	C	Shall be present if a transfer coding has been applied. Shall be absent otherwise.

### 6.X.3.3.3.3 Response Payload

255 If the origin server failed to store or modified any representations in the request payload, the response payload shall contain a Status Report describing any additions, modifications, or deletions to the stored representations.

The Status Report may also describe any warnings or other useful information.

## 6.X.3.4 Search Transaction

260 The Search transaction searches the collection of NPI Instances contained in the target resource. The search criteria are specified in the query parameters. Each match includes the default and requested attributes from the matching Instance. A successful response returns the matching Instances.

### 6.X.3.4.1 Request

The Search service uses the GET method and has the following syntax:

```
265 GET SP /{npi-name} {?parameter*} SP version CRLF
Accept: 1#dicom-media-type CRLF
*(header-field CRLF)
CRLF
```

### 6.X.3.4.1.1 Resources

The target URI shall reference the All Instances resource. An origin server that supports the Search transaction shall support the resources specified in Table 6.X.3-9.

### 270 6.X.3.4.1.2 Query Parameters

The parameters in the query component of the target URL specify the matching criteria, the attribute values to be returned, and the results to be returned. The URI template for the query parameters is:

```
{?parameter*} = "?" {&match*} {&include*} {&offset} {&limit}
```

See Section 6.7.1.1 for a description of the syntax of Search Query Parameters.

### 275 6.X.3.4.1.2.1 Attributes and Behaviors

For each Resource Category, the origin server supports, it shall support the behaviors and matching key attributes as specified in the corresponding Information Model sections in Table 6.X.3-13.

**Table 6.X.3-13: NPI Resource Search Attributes**

Resource Category	Information Model Sections
Color Palette	PS3.4, Annex X.1.3
Defined Procedure Protocol	PS3.4, Annex BB.6.1
Hanging Protocol	PS3.4, Annex U.1.3
Implant Template	PS3.4, Annex BB.1.3

### 6.X.3.4.1.3 Request Header Fields

280

**Table 6.X.3-14: Search Request Header Fields**

Header Field	Value	Usage
Accept	dicom-media-type	M

**6.X.3.4.1.4 Request Payload**

The request has no payload.

**6.X.3.4.2 Behavior**

285

The origin server shall perform the search indicated by the request, using the matching behavior specified in Section 6.7.1.2.1 and in the corresponding Information Model sections in Table 6.X.3-13, and return a response containing the search results, or an appropriate Status Report.

The paging rules are specified in Section 6.7.1.2.1.

**6.X.3.4.3 Response**

290

A success response shall have a status code of 200 (OK) and a payload containing the search results in the Selected Media Type.

A failure response shall contain a Status Report describing the error(s) encountered.

**6.X.3.4.3.1 Status Codes**

The response shall have an appropriate status code. Table 6.X.3-14 contains the most common status codes for this transaction.

295

**Table 6.X.3-14: Common Status Codes**

Status Code	Description
200 (OK)	Indicates that the origin server found at least one resource matching the request.
400 (Bad Request)	Indicates that the origin server did not return any search results because of errors in the request message.
404 (Not Found)	Indicates that the origin server did not find any resources matching the request, or is not willing to disclose that one exists.
406 (Unsupported Media Type)	Indicates that the origin server does not support any of the Acceptable Media Types.
409 (Conflict)	Indicates that the request could not be completed due to a conflict with the current state of the target resource.

**6.X.3.4.3.2 Response Header Fields****Table 6.X.3.15: Search Response Header Fields**

ggHeader Field	Value	Usage	Requirement
Content-Type	dicom-media-type	M	
Content-Length	uint	C	Shall be present if no transfer coding has been applied. Shall be absent otherwise.
Transfer-Encoding	encoding	C	Shall be present if a transfer coding has been applied. Shall be absent otherwise.

**6.X.3.4.3.3 Response Payload**

A success response payload shall contain the Search results.

300

A failure response payload shall contain a Status Report describing any failures, warnings or other useful information.

**6.X.4 Media Types**

The origin server shall support the media types listed as Default or Required in Table 6.X.4-1 for all NPI transactions.

**Table 6.X.4-1: Default, Required, and Optional Media Types**

Media Type	Usage
application/dicom	Required
application/dicom+json	Default
multipart/related; type="application/dicom+xml"	Optional

## 6.X.5 Conformance

305 The origin server shall support the transactions listed as Required in Table 6.X.5-1.

**Table 6.X.5-1 Required and Optional Transactions**

Transaction	Support	Section
Retrieve Capabilities	Required	6.X.3.1
Retrieve	Required	6.X.3.2
Store	Optional	6.X.3.3
Search	Required	6.X.3.4

Implementations shall specify in their Conformance Statement (see PS3.2) and the Capabilities Description (see Section 6.8.1.2):

- The implementations role: origin server, user agent, or both
- The supported resources (IODs) for each role

In addition, for each supported transaction they shall specify:

- The supported Query Parameters, including optional attributes, if any.
- The supported DICOM Media Types
- The supported character sets (if other than UTF-8)
- Any additional behaviors related to the transaction

315