Digital Imaging and Communications in Medicine (DICOM)

Supplement 256: Enabling Live Streaming in DICOMweb

Prepared by:

DICOM Standards Committee, Working Group 27

1812 N. Moore St, Suite 2200

Arlington, VA 22209, USA

Status: June 2026, First Read

Developed pursuant to DICOM Work Item 2025-05-B

Table of Contents

Document History

2026.06	Version 00	JM	Initial version with proposed approach, document structure and content.

Open Issues

1	**Issue:** **Context**: **Proposal**: . **Decision:** [WG06-yyyymmdd]

Closed Issues

Scope and Field of Application

This supplement adds functionality to DICOMweb which enables live streaming to DICOMweb in a standardized way.

Changes to NEMA Standards Publications PS 3.18

Create a new Section on the Live Streaming Service and Resources as indicated below

X Live Streaming Service and Resources

X.1 Overview

The Live Streaming Service enables a user agent to manage selective subscriptions to live streams and to perform push streaming. [Compare to Supplement 202, PS3.22, which is quite different.]

In the context of this Service, live streaming refers to communication between two DICOMweb systems in which producers continuously and immediately transmit applicable newly created Instances to subscribed consumers, where the roles of producer and consumer are application roles that are independent of the HTTP roles of user agent and origin server. The concept of live streaming is illustrated in Figure X.1-1.

Kommentiert [JM1]: WG06: Entire instances, what about multi-frame? Can these be partial / chunked? Incremental. Multipart/chunked.

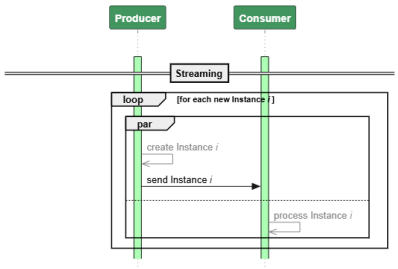

Figure X.1-1. The Concept of Live Streaming

This Service supports two streaming modes: push streaming and pull streaming.

- In **push streaming**, the system acting as the producer assumes the role of user agent and immediately transmits representations of applicable newly created Instances in the payload of a request to a target resource exposed by the receiving system, which acts as the origin server. This transmission is performed using the Live Streaming Service's Add Transaction (see Section X.6).
- In **pull streaming**, the system acting as the consumer assumes the role of user agent and retrieves representations of applicable Instances from a target resource exposed by the producing system, which acts as the origin server, immediately after being notified about their existence. Retrieval is performed using the Study Service's Retrieve Transaction (see Section 10.4). The origin server returns representations of the requested resources in the response payload.

The typical communication and HTTP roles in pull and push streaming are shown in Figure X.1-2.

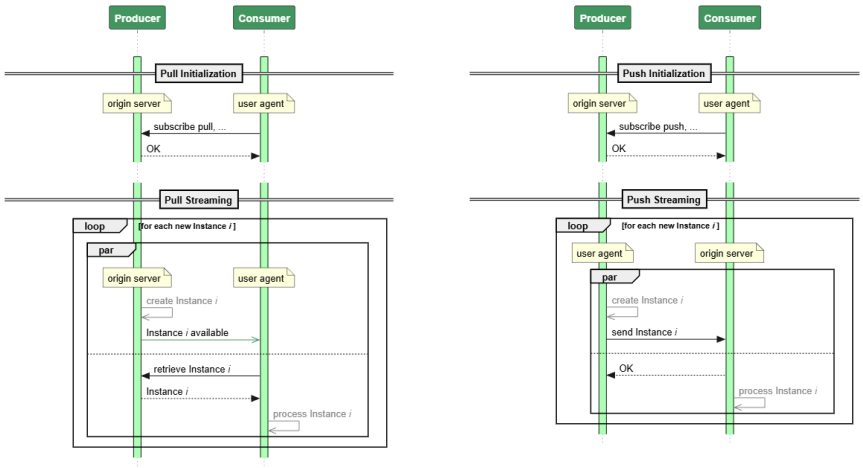

Figure X.1-2. Typical Communication and HTTP Roles in Pull and Push Streaming

In Figure X.1-2, requests are shown as plain arrows, responses as dashed arrows, and notifications as green open-headed arrows. Gray arrows denote internal behavior, not specified in this Section, but required for the context.

~~**UML sequence diagrams as shown in Figure X.1-2 cannot show that messages should be sent as soon as possible.**~~ It is the intent that in pull streaming a) notifications are sent immediately after an instance has been created, b) retrieval is requested as soon as possible after the notification has been received, and c) the retrieval response is sent instantly. For push streaming it is the intent that the Instance is sent once it has been created.

Kommentiert [JM2]: Is expected

[Add some text about security, as this service will give rise to transactions later, not immediately. The server cannot cannot check at transaction time whether there is an access control problem.]

X.1.1 Resource Descriptions

The Live Streaming Service defines the following resources:

Table X.1.1-1. Live Streaming Resources and Descriptions

Resource	URI Template	Description
Stream Subscriptions	/stream-subscriptions/{streamUID}	The stream subscriptions managed by the origin server
Streams	/streams/{streamUID}	The destination of the applicable Instances

Kommentiert [JM3]: Two models: the other one is /streams/{streamUID}/subscription/{...UID}. Rename the current one to /stream-subscriptions/? Also look at SDC/SDPi for patterns. /producers/{...UID}/subscriptions/ when there is a DICOM router or so.

Note The second resource is only relevant for push streaming; it lives at the consumer.

X.1.2 Common Query Parameters

TBS

X.1.3 Common Media Types

TBS; probably including multipart/related

X.2 Conformance

Implementations conforming to the Live Streaming Service shall support the Transactions listed as Required in Table X.2-1.

Table X.2-1. Required and Optional Transactions

Transaction	Support	Section
Subscribe	Required	X.4
Unsubscribe	Required	X.5
Add	Optional	X.6

Implementations conforming to the Live Streaming Service shall specify their role in their Conformance Statement (see PS3.2): origin server, user agent or both.

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).

An origin server conforming to the Live Streaming Service shall implement the Retrieve Capabilities Transaction, specifying its role (see Section 8.9 and Annex H).

An implementation shall, per supported role, specify in its conformance statement whether it supports pull streaming, push streaming, or both. If an implementation claims conformance to push streaming in a role, it shall claim conformance to the Add Transaction for that role; see Section X.6. If an implementation claims conformance to pull streaming in a role, it shall claim conformance to the Study Service's Retrieve Transaction for that role, see Section 10.4; claimed conformance to the latter Transaction shall include the Instance resource, see Section 10.4.1.1.1.

An origin server shall specify in its conformance statement what subscription parameters it supports, even if that is no more than the required matching attributes as specified in Section X.4.1.2.2.

X.3 Transactions Overview

The Live Streaming Service consists of the transactions listed in Table X.3-1.

Table X.3-1. Live Streaming Service Transactions

Transaction Name	Method	Payload		Description
		Request	Success Response	
Subscribe	POST	none	none	Creates a subscription to the streaming of new Instances or the notification of new Instance creation.

Transaction Name	Method	Payload		Description
		Request	Success Response	
Unsubscribe	DELETE	none	none	Cancels a subscription to the streaming of new Instances or the notification of new Instance creation.
Add	POST	Instance	none	Adds a new Instance to the stream of the subscriber.

In Table X.3-2, the target Resources permitted for each Transaction are marked with M if support is mandatory for the origin server, C if it is conditional, and O if it is optional. A blank cell indicates that the resource is not allowed in the Transaction.

Table X.3-2. Resources by Transaction

Resource	Subscribe	Unsubscribe	Add
Stream Subscriptions	M	M	
Streams			C

X.4 Subscribe Transaction

This Transaction uses the POST method to create a Stream Subscription to Instances fulfilling certain characteristics.

When subscribing for push streaming, it subscribes to the streaming of new Instances that have these characteristics.

When subscribing for pull streaming, it subscribes to the notification of creation of new Instances having these characteristics.

Once a Stream Subscription has been created, the user agent either

- for **push streaming**, will receive Instances with the specified characteristics once they are created – through use of the Add Transaction of this Service, see section X.6, or
- for **pull streaming**, will receive notifications about Instances with the specified characteristics once they are created; after being notified, the user agent is expected to immediately retrieve such instances using the Study Service's Retrieve Transaction, see Section 10.4.

To receive notifications generated by Stream Subscriptions, the user agent must first open a Notification Connection between itself and the origin server using the Open Notification Connection Transaction; see Section 8.10.4.

X.4.1 Request

The request shall have the following syntax:

```
POST SP "/stream-subscriptions/"{streamUID} ?{kind} [&{host}] {&search*} SP version CRLF
[Accept: 1#stream-media-type CRLF]
*(header-field CRLF)
CRLF
```

Kommentiert [JM4]: Do we want to add ways to do frequency subsetting, e.g. every second Instance? This would reduce unwanted transfer of Instances when not needed. Of course, the consumer is to decide on that. Frequency on periods too.

Kommentiert [JM4R2]: kind => mode

where

{kind} = "kind" "=" ("push" / "pull")

and {host} is the value to be used in the Host header field of applicable Add Transactions. This value is required when the value of kind equals push.

X.4.1.1 Target Resources

The origin server shall support the resources in Table X.4.1-1.

Table X.4.1-1. Subscribe Transaction Resources

Resource	URI Template
Stream Subscriptions	/streams/subscriptions/{streamUID}

X.4.1.2 Query Parameters

The origin server shall support Query Parameters as required in Table 8.3.4-1; however, the includefield, limit, and offset parameters are ignored.

The user agent shall supply Query Parameters as required in Table 8.3.4-1; however, the includefield, limit, and offset parameter are ignored.

X.4.1.2.1 Attribute/Value Pair Requirements

DICOM Attribute/Value pairs included as Query Parameters in the request shall satisfy the requirements in Section 8.3.4.1.

The user agent may include the following Attributes in the request:

- Patient IE Attributes
- Study IE Attributes
- Series IE Attributes
- Composite Instance IE Attributes
- Private Data Element Tags and their corresponding Private Creator Element Tags
- Timezone Offset From UTC (0008,0201)

Note The Timezone Offset From UTC Attribute has a specific meaning. This is explained in Section 8.3.4.1.1.

The following are examples of Query Parameters with valid Attribute/Value pairs:

- PatientID=11235813&StudyDate=20270509-20270510
- 00100010=SMITH*&00101002.00100020=11235813
- StudyInstanceUID=1.2.392.200036.9116.2.2.2.2163.1926.94587,1.2.392.2036.9116.2.2.2.216289.1026.94583
- 00230010=AcmeCompany
- 00230010=AcmeCompany&00231001=001239
- BodyPartExamined=SHOULDER,NECK

X.4.1.2.2 Required Matching Attributes

The origin server shall support the matching Attributes specified in Table 10.6.1-5.

X.4.1.3 Request Header Fields

The origin server shall support request header fields as required in Table X.4.1-3.

Kommentiert [JM5]: Have a look at what is already there in PS3.18 and mimic that style.

Kommentiert [JM6]: Add mode and host too.

Kommentiert [JM7]: Consider whether we should refactor that out of the common into other services?

Kommentiert [JM8]: Seems a good list to start with.

The user agent shall supply request header fields as required in Table X.4.1-3.

Table X.4.1-3. Request Header Fields

Name	Values	Usage		Description
		User Agent	Origin Server	
Accept	media-type	C	C	The Acceptable Media Types of the Instances in the stream. Required when kind=push.

Kommentiert [JM9]: Or should this be required always? With the current approach, push streaming will detect media type incompatibility when a subscription is created. For pull streaming it is detected when Instances are pulled, which gives the consumer a wrong impression.

See also Section 8.4.

X.4.1.4 Request Payload

The request shall have no payload.

X.4.2 Behavior

When the origin server can create Instances fulfilling the supplied characteristics, it will create the subscription identified by the supplied Stream UID.

X.4.3 Response

The response shall have the following syntax:

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

X.4.3.1 Status Codes

Table X.4.3.1-1 shows some common status codes corresponding to this Transaction. See also Section 8.5 for additional status codes.

Table X.4.3.1-1. Status Code Meaning

Status	Code	Meaning
Success	200 (OK)	The origin server created the Stream Subscription with the supplied Stream UID.
Failure	400 (Bad Request)	The origin server cannot handle the Subscribe request because of errors in the request headers or parameters.
	406 (Unsupported Media Type)	The origin server does not support any of the Acceptable Media Types.
	409 (Conflict)	The origin server cannot handle the Stream Subscription request because the supplied Stream UID is already in use.
	503 (Service Unavailable)	The origin server cannot handle the Subscribe request; this may be a temporary or permanent state.

Note A 400 response may also indicate that one or more parameters have been supplied that the origin server does not support.

X.4.3.2 Response Header Fields

The origin server shall support header fields as required in Table X.4.3.2-1.

163 **Table X.4.3.2-1. Response Header Fields**

Name	Values	Origin Server Usage	Description
Content-Type	media-type	C	See section 8.4.2.
Content-Encoding	encoding	C	See section 8.4.2.
Content-Length	uint	C	See section 8.4.3.

164

165 **X.4.3.3 Response Payload**

166 A success response shall have no payload.

167 A failure response may contain a Status Report describing any failures, warnings, or other useful infor-
168 mation.

X.5 Unsubscribe Transaction

This Transaction uses the DELETE method to delete the Stream Subscription identified by a supplied Stream UID. Such Stream Subscription has been created before as the result of an earlier Subscribe Transaction, see Section X.4.

X.5.1 Request

The request shall have the following syntax:

```
DELETE SP "/stream-subscriptions/"{streamUID} SP version CRLF
*(header-field CRLF)
CRLF
```

X.5.1.1 Target Resource

The Target Resource of this Transaction is a Stream Subscription identified by the supplied Stream UID. The applicable resources are listed in Table X.4.1-1.

X.5.1.2 Query Parameters

The request has no Query Parameters.

X.5.1.3 Request Header Fields

See Section 8.4.

X.5.1.4 Request Payload

The request shall have no payload.

X.5.2 Behavior

The origin server shall delete the Stream Subscription identified by the provided Stream UID.

X.5.3 Response

The response shall have the following syntax:

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

X.5.3.1 Status Codes

Table X.5.3-1 shows some common status codes corresponding to this Transaction. See also Section 8.5 for additional status codes.

Table X.5.3-1. Status Code Meaning

Status	Code	Meaning
Success	200 (OK)	The origin server deleted the Stream Subscription identified by the supplied Stream UID.
Failure	404 (Not Found)	The origin server cannot find the Stream Subscription identified by the supplied Stream UID.
	503 (Service Unavailable)	The origin server cannot handle the Unsubscribe request; this may be a temporary or permanent state.

Note

1. The 404 (Not Found) status code may be caused by an incorrect Stream UID that has been supplied by the user agent, or the origin server may have deleted the applicable Stream Subscription.

2. When the 503 (Service Unavailable) status code is returned, the user agent might retry later with another Unsubscribe Transaction.

X.5.3.2 Response Header Fields

The Response Header Fields are the same as for the Subscribe Transaction. See Section X.4.3.2.

X.5.3.3 Response Payload

The Response Payload is the same as for the Subscribe Transaction. See Section X.4.3.3.

X.6 Add Transaction

This Transaction uses the POST method to enable a user agent to add Instances to the Stream identified by the supplied Stream UID. This Transaction is only used in push streaming, where the producer acts as a user agent, adding Instances to a Stream of an origin server.

Clarify the use of multipart messages. Do we want to allow them? It would reduce the number of HTTP acknowledgements. If so, what are the consequences (1. they shall be chunked! 2. …). How to deal with timeouts in the case of multipart messages? Would an approach be to allow for multipart messages, but when creation is paused, the message shall be ended, or so?

X.6.1 Request

The request shall have the following syntax:

POST SP "/streams/"{streamUID} SP version CRLF
Content-Type: dicom-media-type CRLF
(Content-Length: uint / Transfer-Encoding: encoding) CRLF
*(header-field CRLF)
CRLF
payload

X.6.1.1 Target Resource

The Target Resource of this Transaction is a Stream identified by its Stream UID. The applicable resources are listed in Table X.6.1.1-1.

X.6.1.2 Query Parameters

The request has no Query Parameters.

X.6.1.3 Request Header Fields

See Section 8.4.

The mandatory Host Header Field shall have the host value that has been supplied when creating the Stream Subscription identified by the supplied Stream UID; see Section X.4.

X.6.1.4 Request Payload

The request shall have a payload.

X.6.2 Behavior

The origin server adds the Instance to the Stream identified by the supplied Stream UID.

X.6.3 Response

The response shall have the following syntax:

version SP status-code SP reason-phrase CRLF
CRLF
[payload]

X.6.3.1 Status Codes

Table X.6.3-1 shows some common status codes corresponding to this Transaction. See also Section 8.5 for additional status codes.

Table X.6.3-1. Status Code Meaning

Status	Code	Meaning
Success	200 (OK)	The origin server acknowledges receipt of the Instance.

Failure	400 (Bad Request)	The origin server was unable to receive the Instance due to bad syntax.
	404 (Not Found)	The origin server cannot find the Stream identified by the supplied Stream UID.
	503 (Service Unavailable)	The origin server cannot handle the Add request; this may be a temporary or permanent state.

Note

1. The 404 (Not Found) status code may be caused by an incorrect Stream UID that has been supplied by the user agent, or the origin server may have deleted the stream identified by the supplied Stream UID.
2. When the 503 (Service Unavailable) status code is returned, the user agent might retry later with another Check Send Result Transaction.

X.6.3.2 Response Header Fields

The origin server shall support header fields as required in Table X.6.3.2-1.

Table X.6.3.2-1. Response Header Fields

Name	Values	Origin Server Usage	Description
Content-Type	media-type	C	See section 8.4.2.
Content-Encoding	encoding	C	See section 8.4.2.
Content-Length	uint	C	See section 8.4.3.

X.6.3.3 Response Payload

The Response Payload is the same as for the Send Transaction. See Section X.4.3.3. In a successful response the Status attribute of the payload will be Cancel.

X.7 Instance Created Report

The origin server uses the Send Event Report Transaction (see Section 8.10.5) to send an Instance Created Report, containing the details of the newly created Instance to the user agent.

The origin server shall send Instance Created Reports as described in Table X.7-1.

Table X.7-1. Report Streaming Event – Event Report Information

Event Type Name	Event Type ID	Attribute Name	Tag	Usage (origin server)
Instance Created	1	Study Instance UID	(0020,000D)	1
		Series Instance UID	(0020,000E)	1
		Referenced SOP Instance UID	(0008,1155)	1

The following is an example application/dicom+json Instance Creation Report payload:

```
{ "00001002": {"vr": "US", "Value": [1] }
, "0020000D": {"vr": "UI", "Value": ["..."] }
, "0020000E": {"vr": "UI", "Value": ["..."] }
, "00081155": {"vr": "UI", "Value": ["..."] }
}
```

Kommentiert [JM10]: This table has been formatted as Table CC.2.4-1 in PS3.4. It seemed the best approach to follow that template, even though that is deep down in DIMSE ...

Kommentiert [JM11]: The applicable SOP Instance UID is now explicitly mentioned. Another option would be to use the Affected SOP Instance UID (0000,1000) of the N-EVENT-REPORT Parameters. What would be preferred?

Note While this report requires an Event Type ID, namely that of 1, it does not conform to the N-EVENT-REPORT parameters as specified in PS3.7, Section 10.1.1.1. This would, for instance, require a new SOP Class UID to be created for a non-DIMSE Service, to be used in the Affected SOP Class UID parameter.

Do we want to support multipart here too? This would be possible for a Series Creation Started Report or so. Requires Study Instance UID and Series Instance UID.

Update Section B Examples: add new examples for the respective Streaming Flows

B Examples (Informative)

…

B.X1 TBS

TBS

Update Table H-1 Resources and Methods: add new resources and methods for Send Transactions

H Capabilities Description

Service	Resource	Transactions	Reference
Study (see Section 10.1.1)			
…			
Modality Performed Procedure Steps (see Section 15.1.1)			
	modality-performed-procedure-steps	Create	Section 15.4
		Update	Section 15.5
		Retrieve	Section 15.6
Live Streaming (see Section X.1.1)			
	stream-subscriptions/{StreamUID}	**Subscribe** **Unsubscribe**	**Section X.4** **Section X.5**
	stream/{StreamUID}	**Add**	**Section X.6**

Changes to NEMA Standards Publications PS 3.2

Adapt Annex N of PS3.2

N.1 Overview

…

N.1.3 DICOM Web Services

…

TBS

298 ## Changes to NEMA Standards Publications PS 3.6

299 ***No new attributes have been introduced.***

300

301 ## Changes to NEMA Standards Publications PS 3.15

302 ***There are no new attributes to be added to table E.1-1 of annex E.***

303