Supplement 193

DICOM WEB SERVICE NOTIFICATIONS

WG-27

PUBLIC COMMENT

JIM PHILBIN

2019-06-03

Overview

- Open/Close Notification Connection

- Create/Delete Subscriptions on resources

- Create/Delete Subscription Generators on resources

- Send Event Report

  - Support for Dicom Media Types (CP1871)

  - Now include response message over WebSocket

  - More detailed description of Data Frame format

Goals

- Allows an origin server to send Event Reports asynchronously to a Client

- An Event Report can be

  - Service *independent: system level events that are relevant for all Web Servers. See PS3.4 and PS3.7.*

  - Service *dependent:* Event Reports defined in PS3.28.

- If a Service supports notifications it must

  - Define Event Report Types

  - Define resources that may have Subscription and/or Subscription Generators.

- Allow a Client to create/delete a Subscription Generator to a resource.

Current Standard with CP1871 (VP) included

- Provides a means for a Client to open/close a Notification Connection with an origin server.

    - Asynchronous WebSocket connection

    - Allow the Client to specify the media type of the Event Reports

    - Either Client or origin server may close Notification Connection

    - Provide a reason (Close Code) for closing the connection

Term	Definition
Event	An Event is a change in the content/state of a resource. (trigger)
Event Type	An Event Type is a change in the state of a resource defined in the Standard that has a corresponding Event Report. Each Service defines the Event Types and corresponding Event Reports that it supports, e.g. the UPS Assigned Event Type occurs when the value of Scheduled Station Name Code Sequence or Human Performer Code Sequence is populated. Note: currently the Storage Commit types don't fit with these Events.
Event Report	An Event Report is a DICOM Dataset that is contained in the payload of a Send Event Report Request.
Notification	A Notification is an Event Report message. (maybe drop this definition)
Subscriber	A Subscriber is a user agent that has created a Subscription to a resource on an origin server.
Subscription	A Subscription is an agreement that an origin server will notify a user agent of Events related to a particular resource. The Subscription resource is referenced by an *opaque* URL.
Subscription Generator	A Subscription Generator creates subscriptions, on behalf of a Subscriber, for example when new resources are created (each new sub-resource created). A Subscription Generator is referenced by a *opaque* URL.

Transactions

- Open Notification Connection

- Close Notification Connection

- **Create Subscription**

- **Create Subscription Generator**

  This could be merged into Subscription with a Query Parameter

- **Delete Subscription**

  Deletes either Subscription or Subscription Generator

  Delete Subscription Generator

- **Send Event Report**

Adding Notifications to a Service

If a Service supports notifications it must:

- Define resources that may have Subscriptions and/or Subscription Generators.

- Define Event Types with corresponding Event Reports for resources that may have subscriptions.

Open Notification Connection (CP1871)

Request

```
GET SP / SP version CRLF
Host: host CRLF
Accept: dicom-media-range
Upgrade: "WebSocket" CRLF
Connection: "Upgrade" CRLF
Origin: url CRLF
Sec-WebSocket-Key: nonce CRLF
Sec-WebSocket-Protocol: protocols CRLF
Sec-WebSocket-Version: "13" CRLF
*(<header-field> CRLF)

CRLF
```

Response

```
version SP status-code SP reason-phrase CRLF
Upgrade: "WebSocket" CRLF
Connection: "Upgrade" CRLF
Sec-WebSocket-Accept: response-key CRLF
Sec-WebSocket-Protocol: protocol CRLF
Content-Type: dicom-media-type CRLF
*(header-field CRLF)
CRLF
```

Changes to Subscriptions

- Differentiate between Subscriptions and Subscription Generators

  This allows:

  - Creating a Subscription, e.g. to a service (/) or for example to a workitem (/workitems)

  - Creating a Subscription Generator for a collection resource

- A Subscription or Subscription Generator is a resource that is referenced by an *opaque* URL that is returned in the Location header field of response.

  - This implies a CP to the Worklist Service(UPS)

- The Target Resource should be one of the standard resources defined by the service, i.e. /workitem/{resource}

  The current UPS resource /workitem/{resource}/subscribers/{aetitle} would change to /workitem/{resource}/generator|subscription

- No Global UIDs for Global / Filtered Subscription Generators

  - Global: target URL is /worklist

- To receive System & Service level events create a Subscription to /.

Create Subscription Request

Request
```
POST SP /{/resource}/subscription{?accept}{&deletionLock} SP
version CRLF
Accept: media-types CRLF
*(header-field CRLF)
CRLF
```
Where

{/resource} is defined by the Service.

{&deletionLock} if true the subscription is not deleted until the lock is released.

- *The resource should be one of the standard resources defined by the service.*
  The current UPS resource /workitem/{resource}/subscribers/{aetitle} becomes /workitem/{resource}/subscription
- *Should simple subscriptions have a deletion lock?*
- *A simple subscription may not have a filter – should it? It could specify the types of event notifications requested*

Create Subscription Response

```
Response
    version SP status-code  SP reason-phrase CRLF
    Location: subscription CRLF
    *(header-field CRLF)
    CRLF
    [status-details]
```

* subscription is an opaque URL.

Create Subscription Generator Request

Request

```
POST SP /{/resource}/generator{?accept}{&deletionLock}{filter} SP
version CRLF
Accept: media-types CRLF
*(header-field CRLF)
CRLF
```

Where

{/resource}   is defined by the Service.
{&deletionLock} if true the subscription is not deleted until the lock is released.
{&filter}   one or more matching keys

- The Target Resource must be a collection resource, i.e. contain sub-resources.
- *The Target Resource should be one of the standard resources defined by the service.*
  The current UPS resource /workitem/{resource}/subscribers/{aetitle} becomes /workitem/{resource}/generator

Create Subscription Generator Response

```
Response
    version SP status-code  SP reason-phrase CRLF
    Location: subscription CRLF
    *(header-field CRLF)
    CRLF
    [status-details]
```

* subscription is an opaque URL.

Event Reports

Event Reports must support both data frames and text frames in order to support deferent media types.

- *The Dataset is encoded in the format of the media-type in the Content-Type header field of the Open Notification Connection response.*
- *The elements required/optional in all Event Reports needs to be better specified. What about (0000,xxxx) message elements? Should they be included?*
- *Specific Event Report element requirements should be specified by the Service by Event Type.*

Send Event Report Request

Request

```
request = frame-header frame-length dataset
```

Where

```
frame-header = text-frame / data-frame
 text-frame   = %b100000010 ; 9-bit origin server frame header
 data-frame   = %b100000100 ; 9-bit origin server frame header
frame-length = ; 7, 23, or 71-bit encoded length - see RFC6455
```
dataset   = +octet ; dataset is encoded in a DICOM Media Type

The Dataset is encoded in the format of the media-type in the Content-Type header field of the Open Notification Connection response.

Send Event Report Response

Response

```
response = frame-header frame-length (ack/nack) [utf8]
```

Where

```
frame-header  = text-frame / data-frame
 text-frame   = %b100000011 ; 9 bits
 data-frame   = %b100000101 ; 9 bits
payload-length               ; see RFC6455 - 7, 23, or 71 bits
Masking-key   = 4octet       ; 32-bit mask for Client-side message
Ack           = %x01         ; 1 byte
Nack          = %x02         ; 1 byte
Utf8          = *octet       ;
```

*utf8 is a UTF-8 encoded octet stream

Delete Subscription

Request

```
DELETE SP {/resource} SP version CRLF
Accept: media-type CRLF
*(header-field CRLF)
CRLF
```
Where
```
{/resource} is the opaque URL returned by Create Subscription
```

Response

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

Delete Subscription Generator

Request

```
DELETE SP {/resource}{?existing} SP version CRLF
Accept: media-type CRLF
*(header-field CRLF)
CRLF
```

Where

```
{/resource} is the opaque URL returned by Create Subscription Generator
{?existing} = "existing" "=" ("keep" / "delete")
```

Response

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

Open Issues

- What notifications should be defined for the Studies service?
    - *Notification that a Study, Series, or Instances has changed (added, changed, archived, deleted)?*
    - *Notification that an Instance has changed (updated/corrected, deleted)?*
- What *system level notifications should be supported?*
    - *Add system notifications to Capabilities Report?*
- Event Report notification/responses need to be better defined?

Information that needs to be included in the PPT

- Be sure to identify target audiences

  - <wg-06 FOR NOW> <NEXT STEP, REAL AUDIENCE>

- Make sure to educate newer people to DICOM about the benefit of using this supplemen

  - t<USE CASES FROM THE WIP>

- Why would potential venders be interested in implementing this supplement, and make a use of "use cases"?

- What is the benefit for the users?

- Please remember to facilitate the readers' understanding of this new proposed addition to the DICOM Standard, the authors who prepare a presentation, need to try to describe in a simple language the technical issues and use cases.