

DICOM Correction Proposal

STATUS	Assigned
Date of Last Update	2019/06/03
Person Assigned	Jim Philbin <jim@philbin.us>
Submitter Name	Jim Philbin <jim@philbin.us>
Submission Date	2018/11/08

Correction Number	CP-1871
Log Summary: Improve Notification Connections	
Name of Standard PS3.18 2016a	
Rationale for Correction: The following need to be improved: <ul style="list-style-type: none">• Allow Open WebSocket Connection to have a media type using the Accept/Content-Type headers in the request/response.• Add Close Notification Connection Transaction	
Correction Wording:	

Update PS3.18, Section 8.10.3 as follows:

8.10.3 Transactions Overview

Any service that supports the Notifications must support the following transactions:

Table 8.10.3-1: Notification Sub-System Transactions

Name	<u>Protocol</u>	Method	Description
Open Notification Connection	<u>http</u>	GET	The user agent requests that the origin server create a Notification Connection between them.
<u>Close Notification Connection</u>	<u>ws</u>	<u>N/A</u>	<u>Closes the Notification Connection. The user agent or the origin server may close the connection.</u>
Send Event Report	<u>ws</u>	<u>N/A</u>	The origin server sends an Event Report to an End-Point

Where ws is the WebSocket protocol.

Update PS3.18, Section 8.10.4.1 as follows:

8.10.4.1 Request

There is more than one way to establish a WebSocket connection. An origin server that conforms to [RFC6455] will at least support requests to open a WebSocket over an HTTP connection that have the following syntax:

```
GET SP / SP version CRLF
Host: host CRLF
Accept: 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

```

The origin server may support other methods of opening a WebSocket connection, which should be included in the Conformance Statement and the Retrieve Capabilities response.

Update PS3.18, Section 8.10.4.1.3 as follows:

8.10.4.1.3 Request Header Fields

Table 8.10.4-1 shows the Request Header Field usage for opening a WebSocket connection over http/https.

Table 8.10.4-1. Request Header Fields

Name	Value	Usage		Description
		User Agent	Origin Server	
<u>Host</u>	<u>url</u>	<u>M</u>	<u>M</u>	<u>See [RFC6455]</u>
<u>Content-Type-Accept</u>	<u>dicom-media-types</u>	M	<u>M</u>	<u>Acceptable Media Types for WebSocket messages</u>
Upgrade	"WebSocket"	M	<u>M</u>	<u>See [RFC6455]</u>
Connection	"Upgrade"	M	<u>M</u>	
Origin	url	M	<u>M</u>	
Sec-WebSocket-Key	accept-key	M	<u>M</u>	
Sec-WebSocket-Protocol	protocols	O	<u>O</u>	
Sec-WebSocket-Version	version	M	<u>M</u>	

The Accept header field should specify the Acceptable Media Types for the WebSocket connection.

For details of the request header field values and other methods of opening a WebSocket connection see [RFC6455]

Update PS3.18, Section 8.10.4.2 as follows:

8.10.4.2 Behavior

When the origin server receives this request, it shall open and maintain a WebSocket connection between itself and the user agent.

The origin server shall choose a Selected Media Type from the Accept header field in the request. If the Accept header field is not present the media type for the WebSocket connection shall be application/dicom+json.

If the connection is lost at any point, the user agent can re-establish it by repeating this transaction.

Update PS3.18, Section 8.10.4.3 as follows:

8.10.4.3 Response

The response shall have the following syntax:

```

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

```

Update PS3.18, Table 8.10.4-2 as follows:

Table 8.10.4-2. Status Code Meaning

Status	Code	Meaning
Success	101 (Switching Protocols)	The protocol was successfully changed to WebSocket.
Failure	400 (Bad Request)	There was a problem with the request.
	<u>406 (Not Acceptable)</u>	<u>The origin server does not support any of the Acceptable Media Types. See Section 8.5.</u>

Update PS3.18, Table 8.10.4-3 as follows:

Table 8.10.4-3. Response Header Fields

Name	Value	Origin Server Usage	Description
<u>Content-Type</u>	<u>dicom-media-type</u>	<u>M</u>	<u>The Selected Media Type the Origin server will use on the WebSocket connection.</u>
Upgrade	"WebSocket"	M	<u>See [RFC6455]</u>
Connection	"Upgrade"	M	
Origin	url	M	
Sec-WebSocket-Accept	response-key	M	
Sec-WebSocket-Protocol	protocol	M	

Update PS3.18 Section 8.10.5.1.1 with the following:

8.10.5.1.1 Request Payload

~~The data frames shall have an opcode of "%x1" (text).~~

~~The data frame payload data shall be a DICOM JSON Dataset containing the Attributes of the Event Report.~~

Events Reports are encoded as WebSocket Data Frames. There are two types of Data Frames specified by the Opcode field:

- **%x1 denotes a text frame**
- **%x2 denotes a binary frame**

Notification messages have the following syntax:

frame-header payload-length event-report

Where

frame-header = text-frame / binary-frame

text-frame = %b100000010 ; A 9-bit text Data Frame header

binary-frame = %b100000100 ; A 9-bit binary Data Frame header

payload-length ; 7, 23, or 71-bit encoded length, (see [RFC6455] Section 5.2.

event-report ; A Dataset specified by the defining Service and encoded in the ; Dicom-Media-Type specified when the WebSocket connection was opened.

The Data Frame Type shall be specified by the media type of the Event Report.

The Event Report content, and available media types shall be specified by the Defining Service.

Note:

1. The WebSocket protocol does not currently allow content negotiation so it is not possible to support both XML and JSON encoding of Event Report messages.
2. ~~If the Event Reports are being proxied into DIMSE N-EVENT Reports, a Message ID (0000,0110) must be managed by the proxy.~~
2. **WebSocket frames from a Client to an origin server must be masked. See [RFC6455] Section 10.3.**
3. ~~If the Event Reports are being proxied into DIMSE N-EVENT Reports, a Message ID (0000,0110) must be managed by the proxy.~~

Update PS3.18 Section 8.10.5.3 as follows:

8.10.5.3 Response

~~None.~~

The user agent shall send an acknowledgement response containing a Status Code. The response Status Code can be either general (See PS3.7, Section C) or specific to the service. Each service may define response codes specific to that service, which should be the same codes used by the corresponding DIMSE service, if any. The response is encoded as a WebSocket (binary) data frame with an opcode of "%x2" (binary). See [RFC6455]<<https://tools.ietf.org/html/rfc6455#page-27>>.

8.10.5.3.1 Response Payload

The response payload has the following syntax:

opcode SP ack / nack SP media-type CRLF
[status-report]

Where

frame-header = text-frame / binary-frame
text-frame = %b100000011 ; A text Data Frame with an Opcode of Text (%x1)
binary-frame = %b100000101 ; A binary Data Frame with an Opcode of Binary (%x2)
frame-length ; A 7, 23, or 71-bit encoded length
mask ; A 32-bit mask that is generated randomly
ack = 1 / "success"
nack = 0 / "failure"
media-type ; is an HTTP media type
event-report ; is a Dataset

Insert the following into PS3.18 as Section 8.10.5.4:

8.10.5.4 Media Types

All implementations of this sub-service must support the "application/json" media type. Other required or recommended media types may be specified by the Defining Service.

Table 8.10.5-3. Event Report Media Types and OP Codes

Media Type	OP Code	Frame Type
application/dicom	%x2	binary
application/dicom+json	%x1	text
application/dicom+xml	%x1	text

Insert the following into PS3.18, as Section 8.10.6:

8.10.6 Close Notification Connection Transaction

This transaction closes a connection between the user agent and the origin server. It uses the WebSocket Data Frame transmission protocol (see [RFC6455]) and may be initiated by either the user agent or origin server.

8.10.6.1 Request

The initiator shall use the WebSocket Data Frame transmission protocol.

The Close Code can be one of the codes from the WebSocket Close Code Number Registry<<http://www.iana.org/assignments/websocket/websocket.xhtml>> (1000-2999), one of the DICOM defined codes from Table 8.10.6.1, or an implementation defined Close Code in the range of 4000-4999.

Table 8.10.6-1. Close Notification Connection Reason Codes

Status	Description	Reason Code
Success	The requested change of subscription state was performed	0000
Warning	Deletion Lock not granted.	B301

Failure	Specified SOP Instance UID does not exist or is not a UPS Instance managed by this SCP	C307
	Receiving AE-TITLE is Unknown to this SCP	C308
	Refused: Specified action not appropriate for specified instance	C314
	Refused: SCP does not support Event Reports	C315

8.10.6.2 Behavior

The receiver of the request shall close the connection as specified in [RFC6455] Section 5.5.1.

8.10.6.3 Response

None.

8.10.6.4 Conformance

An implementation supporting the notification transactions shall describe its support in the Conformance Statement, and whether it plays the role of origin server or user agent, or both.

An implementation playing the role of origin server shall also describe the following in its Conformance Statement.