Digital Imaging and Communications in Medicine (DICOM)

DICOMWEB SUPPORT FOR RETRIEVE VIA APPLICATION/ZIP

Prepared by:

Brad Genereaux

DICOM Standards Committee, Working Group 27

1300 N. 17th Street

Rosslyn, Virginia 22209 USA

VERSION: 6

January 7, 2019

Table of Contents

Forward i
Open Questions i
Closed Questions ii
8.3.3.3 Password Query Parameter iv
8.7.3.4 DICOM Media Type Syntax v
8.7.3.8 The application/zip Media Type v
8.7.3.4.2 DICOM Zip Media Type v
8.7.3.4.5 Password Parameter vi

Forward

Supplement 211 purports to add “application/zip” as an option in the DICOMweb WADO-RS specification, to retrieve an entire DICOM study or series as a single packaged file.

The primary use case is to enable researchers in analytics / machine learning users of DICOMweb who want to retrieve studies / series / collections of images for training purposes. DICOMweb currently provides a means to retrieve an entire study or an entire series using a “multipart/related” content type, but this requires special non-browser implementations to retrieve the content (multipart/related is not supported by any browsers; it would require special programmatic interventions). It also provides the objects separately (as opposed to a bundle) making them more difficult to work with on disk / importing into their machine learning tooling as they are streamed into individual files.

A secondary use case is for consumer use of DICOM objects, as providing a ZIP file of DICOM content to a patient or caregiver is more familiar than providing them with a multipart/related response to be programmatically interpreted.

Open Questions

Q	Question	Position
3b	(Brad) Do we want to have parameters to support ZIP encryption?	We believe so - ZIP encryption is useful for data "at rest“ once the client retrieves the ZIP file (it is irrelevant for data "in transfer" as HTTPS encrypts traffic). Encrypting at the client is less efficient than specifying at time of request.
4	(Brad) Do we need to specify any details regarding the transfer syntax / media types of the contained files?	We believe so, using the same structure/methodology as what multipart/related supports
5	(David) Do we want to specify a different RESTful resource (e.g., /packaged)	No, we don’t believe that’s necessary; it’s essentially a transformation on the same path as other WADO-RS services. As with any of the REST services DICOMweb offers, the design tended to follow the principles of REST Uniform Interface (https://en.wikipedia.org/wiki/Representational_state_transfer#Uniform_interface) and Clean URLs

		(https://en.wikipedia.org/wiki/Clean_URL); ZIP is just a transformed representation of a DICOM study and it is acceptable and encouraged by REST community to specify it in this way In line with how other APIs work and is more discoverable and what's expected
9	(Charles) The "README" of the file that indicates "where it came from" and other metadata – extremely important.	Might be something for IHE / PDI, whoever composed the ZIP; see https://www.ihe.net/uploadedFiles/Documents/Radiology/IHE_RAD_TF_Vol3.pdf, in particular section 4.47.4.1.2 to see if anything is worth mirroring, with specifics about the README in 4.47.4.1.2.2.2
10	(Elliot) Are there conceivable limits to the ZIP file size itself? What about DICOMDIR limits? Do we need to limit the # of items in the root directory?	This problem would apply to any implementation of DICOM Zip – solutions would apply beyond web; maybe we should make a note in "notes" of any upper limits that is useful for implementers
11	(Rob) Do we need to update the reference to ZIP as a standard? The reference standard is currently https://support.pkware.com/display/PKZIP/APPNOTE	No, in as far as we can tell, this remains the correct URL to reference
12	(Rob) Are there security concerns about popping a URL into Chrome for example?	We need to think through if there are any concerns; e.g., is there special behavior if someone tries to download but not authenticated
13	(Elliot) What happens if a client accepts both multipart/related and application/zip and are "weighted" equally?	HTTP protocol specifies it is a server's decision to choose then, and we believe that mantra should still apply
15	Should we consider ZIP at the instance level, or just at the study / series level?	(answer depends a bit on solution to question 5)
16	Should we consider ZIP at the rendered level as well?	November 2018 discussions suggested this should be in scope (meets need for algorithm developers with JPG libraries); December 2018 discussions suggested this should be removed from scope (many "gotchas" in what people would need to specify)
17	Do we need a chapter in PS3.12 Appendix V to describe any specific behavior for network transport / on the wire for ZIP files?	

Closed Questions

Q	Question	Notes
1, 2, 7	Is "zip" a "DICOM media type", a rendered media type, or something else? / Do we want to include support for DICOMDIR structures, or make it the default (or only) format? / Would we consider supporting the DICOM media type already defined in Part 12 section V?	Yes, it is a DICOM media type and we will use what is defined in http://dicom.nema.org/medical/dicom/current/output/html/part12.html#chapter_V. We will use existing DICOM Zip specification, which uses DICOMDIR; it supports existing implementations already using DICOM Zip (one implementer confirmed); committee recognizes this may not be "ideal" but the cost doesn't justify the benefit of the change; note recipients have no requirements to do anything with the DICOMDIR
3a	Do we want to have parameters to support anonymization?	No; anonymize not supported by WADO-RS, and not directly related to this supplement and thus will not be included as part of this supplement; recipients can anonymize the data after receipt; or data could be

		pre-anonymized. It just isn't in this step of the pipeline
6	Do we want to allow support for tar or rar, be silent on it, or explicitly disallow it?	The specification describes application/zip; any other packaging formats will not be specified.
8	Can we support retrievals that are completely specified in a URL (and not require Accept header to be passed)?	Yes; this is already supported in PS3.18 6.1.1.5 (query parameter "accept"). http://dicom.nema.org/medical/dicom/current/output/html/part18.html#sect_6.1.1.5
14	Should WG-27 tackle broader DICOM Zip issues, beyond the scope of the web component?	Yes, where possible and what makes sense / is appropriate

Modify PS3.12 Appendix V title and add introduction as indicated.

V ZIP File ~~Media~~ Encoding (Normative)

A DICOM ZIP file containing instances is used both for zipping DICOM part 10 instances as a physical media and also as a container for distributing the DICOM Part 10 files that can be retrieved via network download. See PS3.18 8.7.3.8 for WADO-RS retrieval mechanism via ZIP.

V.1 DICOM Mapping to ZIP File

V.1.1 DICOM File-set

One and only one DICOM File-set shall be contained in a ZIP File archive.
[…]

Add new section PS3.12 V.2.1 after V.2 Logical Format.

V.2.2 Password Encryption

If requested and supported by the Service Class Provider, the ZIP File can be encrypted with a supplied password, which is thus used to have the data secured when stored at rest by the Service Class User.

See PS3.18 8.3.3.3 for the WADO-RS mechanism to specifying the password for encryption.

Add new section PS3.18 8.3.3.3 after 8.3.3.2 Character Set Query Parameter

8.3.3.3 Password Query Parameter

The password Query Parameter is an optional parameter for specifying the encryption key for DICOM ZIP media responses.

It is optional for the origin server. It is optional for the user agent.

The password Query Parameter has the following syntax:

```
password = "password" "=" %s(*octet)
```

The password Query Parameter value is a single value.

If this parameter has a value that is not valid, the origin server shall return a 400 (Bad Request) response and may include a payload containing an appropriate Status Report. See Section 8.6.3.

Modify PS3.18 8.7.3.4 DICOM Media Type Syntax

8.7.3.4 DICOM Media Type Syntax

The syntax of DICOM Media Types is:

```
dicom-media-type = (dcm-singlepart / dcm-multipart) [dcm-parameters]
```

Where

```
dcm-singlepart = dcm-mt-name
dcm-multipart                                    ; see Section 6.1.1.8.1.1
dcm-parameters = transfer-syntax-mtp             ; see Section 6.1.1.8.1.2
               / charset-mtp                     ; see Section 6.1.1.8.1.3
dcm-mt-name       = dicom / dicom-xml / dicom-json  ; DICOM Media Type name
dicom                   = "application/dicom"
dicom-xml         = "application/dicom-+xml"
dicom-json        = "application/dicom+json"
octet-stream      = "application/octet-stream"
zip               = "application/zip"
```

Add PS3.18 8.7.3.8 after 8.7.3.7 Support for DICOM Media Types by Service

8.7.3.8 The application/zip Media Type

The application/zip media type specifies that the representation is a ZIP encoding of a DICOM File Set as specified in PS3.12, Section V.

8.7.3-8. Media Types for DICOM ZIP Files

Media Type	Descriptions	URI	RESTful
application/zip	Encodes DICOM Composite SOP Instances (Part 10) in a ZIP file	Not applicable	See Table 7.10-7

Table 8.7.3-8 specifies the default and optional Transfer Syntax UID combinations for each application/zip Resource Category (see Table 8.7.2-1) for the RESTful service. The default media type for the Resource Category shall be returned when the origin server supports none of the Acceptable Media Types.

If no media type Transfer Syntax parameter is specified, then the Explicit VR Little Endian Transfer Syntax "1.2.840.10008.1.2.1" shall be used.

Note:

This is different from the Default Transfer Syntax defined in PS3.5 Section 10.1, which is Implicit VR Little Endian.

The origin server may support additional Transfer Syntaxes.

Add PS3.18 8.7.3.4.2 after 8.7.3.4.1 DICOM Multipart Media Types

8.7.3.4.2 DICOM Zip Media Type

The syntax of DICOM media types is:

```
dcm-zip  = “application/zip”
```

See Section 8.6.1.2.1 for the definition of related parameters.

Add PS3.18 8.7.3.4.5 after 8.7.3.4.3 Character Set Parameter

8.7.3.4.5 Password Parameter

A DICOM Zip may have a encrypted password specified, which shall have only a single value.

The syntax is:

```
password = password
```

See PS3.12 Appendix V.2.1 for details.