

Digital Imaging and Communications in Medicine (DICOM)

Supplement 234: DICOMweb Storage Commitment

Prepared by: Working Group 27

DICOM Standards Committee, Working Group 6

1300 N. 17th Street, Suite 900

Rosslyn, Virginia 22209 USA

Status: Version 0.2, November 2022, Public Comment Draft

Developed pursuant to DICOM Work Item 2022-04-A

Table of Contents

Document History	5
Open Issues	5
Closed Issues	5
Scope and Field of Application	10
2 Normative References	11
2.3 Other References	11
10 Studies Service and Resources	11
10.1 Overview	11
10.1.1 Resource Descriptions	11
10.1.2 Common Query Parameters	13
10.1.3 Common Media Types	13
10.2 Conformance	13
10.3 Transactions Overview	13
10.x Commit Transaction	15
10.x.1 Request	16
10.x.1.1 Target Resources	16
10.x.1.2 Query Parameters	16
10.x.1.3 Request Header Fields	16
10.x.1.4 Request Payload	17
10.x.2 Behavior	17
10.x.3 Response	18
10.x.3.1 Status Codes	18
10.x.3.2 Response Header Fields	18
10.x.3.3 Response Payload	19
10.x.4 Result Check	21
10.x.4.1 Target Resources	21
10.x.4.2 Query Parameters	22
10.x.4.3 Result Check Header Fields	22
10.x.4.4 Result Check Payload	22
10.x.5 Result	22
10.x.5.1 Status Codes	22
10.x.5.2 Result Header Fields	23
10.x.5.3 Response Payload	23
10.x.6 Media Types	23
10.x.7 Conformance Statement	23
B Examples (Informative)	24
B.x1 Request Commit for Multiple Instances with JSON	24
B.x2 Request Commit for Multiple Instances with XML and Referenced Study and Series Instance UIDs	26
B.x3 Request Commit with HTTP Multipart Request for Multiple Studies	27
H Capabilities Description	29
A DICOM Conformance Statement Template (Normative)	32
A.1 Overview	32
A.1.3 DICOM Web Services	32
A.1.3.2 Study Service	32
A DICOM Conformance Statement Template (Normative)	33
A.5 Service and Interoperability Description	33

A.5.3	Supported DICOM Web Services	33
A.5.3.2	Studies Web Service	33
A.5.3.2.x	Commit Transaction	33
A.5.3.2.x.1	User Agent	33
A.5.3.2.x.2	Origin Server	34
A	DICOM Conformance Statement Template (Normative)	35
A.6	Configuration	35
A.6.3	Configuration of DICOM Web Services	35
A.6.3.2	Studies Web Service	35
A.6.3.2.x	Commit Transaction	35
A	DICOM Conformance Statement Template (Normative)	36
A.7	Network and Media Communication Details	36
A.7.3	Status Codes	36
A.7.3.3	DICOM Web Services	36
A.7.3.3.3	Studies Web Service	36
A.7.3.3.3.7	Commit Transaction as Origin Server	36
A.7.3.3.3.8	Commit Transaction as User Agent	36
6	Registry of DICOM Data Elements	38

Document History

September 2022	Version 0.1	JM	Initial version, ready for First Read (FR) of WG06.
November 2022	Version 0.2	DK & JM	<p>Updated with results from WG06's FR:</p> <ul style="list-style-type: none"> • Closed issues. • Synchronous and asynchronous behavior. <p>Extended to be ready for discussion for Public Comment:</p> <ul style="list-style-type: none"> • Closed a lot of issues. • Specified the entire subsection 10.x. • Added three examples (in Annex B). • Added Commit in Capabilities Description (Annex H). • Added changes to Conformance Statement template (pending the approval of supplement 209), including review comments of WG31's people. • Added changes to DICOM's data dictionary. • Added changes to de-identification profiles.

Open Issues

15	<p>Issue:</p> <p>Proposal:</p> <p>Conclusion:</p>
----	--

Closed Issues

1	<p>Issue: In what section of Part 18 should the storage commit service be put? Is it part of the Studies Service and Resources, section 10, or should it be specified as a separate Service (so as section 13)?</p> <p>Proposal: make this service part of the Studies Service and Resources, as storage commitment (safekeeping of studies) fits the description of the Studies Web Service as given in section 7.1.2: "Enables a user agent to manage Studies stored on an origin server."</p> <p>Decision (2022-09-19 WG-06): Agree with proposal.</p>
2	<p>Issue: How to name the service that handles storage commitment? Is it the Storage Commitment Service? Is it the Commit Storage Service? Is it the Request for Storage Commitment Service? Or is it the ...</p> <p>The first option mentioned resembles the SOP Class best but is not actively put as the other web services (Store, Retrieve, Search). The second one makes it active, with a verb, but suggests that the user agent is committing, while in fact the origin server is committing. The third option describes the service the best, and is active, but is quite long.</p> <p>Proposal: the name of the service will be Commit Storage Service (as the Store service is not called Request to Store service either).</p>

	<p>Decision (2022-09-19 WG06): name the service the Commit service. This is also one word, active, etc. Furthermore, nothing else needs to be committed, so there is no ambiguity.</p>
3	<p>Issue: How to deal with synchronicity? Do we allow for synchronous or asynchronous communication only, or for both? What will be the approach in either case?</p> <p>Proposal: We will allow for both kinds of communications. The starting point of communication will be the Asynchronous Request-Reply Pattern as for instance specified here: Asynchronous Request-Reply Pattern - Azure Architecture Center Microsoft Docs. This means that <i>polling</i>¹ will be the basis of the architecture, which is apparently quite normal in RESTful architectures. This pattern will need to be adapted for this specific use case, e.g. it will be allowed that synchronous communication is possible; see issues 3a-3h for all adaptations.</p> <p>Option: The origin server can return the expected polling frequency to the user agent. The user agent should not poll too often, say at most each 5 minutes, as this will increase the load of the origin server. In each polling reply, the origin server could provide the number of seconds before the next poll, depending on the current load of the server (a kind of expectation management). If a user agent is polling earlier than expected, an easy error status code should be provided.</p> <p>Decision: As proposed.</p>
3a	<p>Issue [WG06]: do we only provide asynchronous or also synchronous response of the origin server?</p> <p>Proposal: Allow for both. Therefore, the solution as provided by DICOMweb will be hybrid, as is DIMSE. This makes that happy flows include both 200 (OK, so done) and 202 (Accepted, so still working) responses to the POST request. The origin server determines the synchronicity of the commit transaction; the user agent needs to follow suit.</p> <p>Decision: As proposed.</p>
3b	<p>Issue: How to convey the (possible) failure codes as defined for storage commitment?</p> <p>Proposal: Provide these in the response payload. HTTP errors are only for that level of communication. This approach implies that the result needs to be checked by the user agent (as is done for DIMSE too).</p> <p>Decision: As proposed.</p>
3c	<p>Issue: What resources are used for requesting storage commitment?</p> <p>Proposal: Only one, the <code>commit</code> resource. See issue 4.</p> <p>Decision: As proposed.</p>
3d	<p>Issue: What different resource is to be used for polling for the result/checking the status of the request?</p> <p>Proposal: No resource at all. It is proposed to simplify the pattern by using GET on the (single) Commit resource. This approach is simple as it requires no other resources (and management of them).</p> <p>Decision: As proposed.</p>
3e	<p>Issue: What resource is being created as status resource?</p> <p>Proposal: No resource will be created as status resource. The HTTP status codes are being used to make the distinction between the different scenarios (ready, (still) working, error). Again, the advantage is not to create resources for single use (or so).</p> <p>Decision: As proposed.</p>

¹ Using a call-back mechanism may be insecure, would require additional implementation effort for the both the user agent and the origin server, and would a handle to be passed to the origin server with which the user agent can be called back.

3f	<p>Issue: What resource is being created as result?</p> <p>Proposal: No resource is being created as result. The result will be given back as the payload of the response. Same advantage as before.</p> <p>Decision: As proposed.</p>
3g	<p>Issue: What kind of HTTP status codes are supported?</p> <p>Proposal: 200 (OK, done with the request, check the result), 202 (still working), 400 (for a bad request), and 404 (when the result of a request with the specified transaction UID cannot be found).</p> <p>Decision: As proposed.</p>
3h	<p>Issue: Do we want to be able to cancel this long running request?</p> <p>Proposal: No as this is also not possible in DIMSE.</p> <p>Decision: As proposed.</p>
4	<p>Issue: Do we have a separate commit storage transaction that takes a list of references SOP Instance UIDs, or do we commit per Instance resource? The first is more in line with the DIMSE way of working (and requires a lot less transactions) while committing per separate SOP Instance fits the resources model best (e.g. <code>/instance/{uid}/commit</code>).</p> <p>Proposal: Keep the DIMSE model: one request with a lot of Instance UIDs, as this has a quite smaller number of transactions compared to separate calls. This is also the way Orthanc works. Having commitment per separate Instance too (as a second way to achieve the same) would be interface overload and is not chosen to be supported.</p> <p>Note that we could – in principle – have commit transactions on study or series level, but that would limit the granularity, and still increase the number of transactions.</p> <p>Decision (2022-09-19 WG06): Agree with proposal. If people want to commit per instance, they still can, using the chosen interface.</p>
5	<p>Issue: Do we need to pass the Referenced SOP Class UIDs with the commit storage request as is done in DIMSE? What is/has been the purpose of that? If there is no clear purpose, we could skip it. However, in that case behavior is different than the DIMSE storage commit. When this cannot be skipped, it makes much more sense to resolve issue 8 with payload instead of parameter.</p> <p>Proposal: Get rid of the Referenced SOP Class UIDs (and the optional Storage Media File-Set (U)ID). However, this would require a new sequence attribute, as the Referenced SOP Class UID is type 1.</p> <p>Decision (2022-09-19 WG06): Do not remove the Referenced SOP Class UID, as this would violate the proxy-ability.</p>
6	<p>Issue: Does the origin server need to pass back all Referenced SOP Instance UIDs that have successfully been committed by the origin server? It is like DIMSE, but what can we do with it (except check whether the origin server is giving back some strange results)?</p> <p>Proposal: Pass back all Referenced SOP Instance UIDs, as it may be easier for the user agent this way (may not need a separate administration).</p> <p>Decision (2022-09-16 WG06): As proposed, also for backwards compatibility.</p>
7	<p>Issue: Does the origin server need to pass back the Referenced SOP Class UIDs? The approach to this item should match that of issue 5.</p> <p>Proposal: No, this is no longer needed. In line with this, the optional Storage Media File-Set (U)ID are no longer possible.</p> <p>Decision (2022-09-19 WG06): keep in line with issue 5, so do return this too.</p>
8	<p>Issue: Shall we pass the list of Referenced SOP Instance UIDs as HTTP parameter or as payload? Parameter seems to fit the current model (as it is also used in querying), but that seems not possible for the response, and would hence be asymmetrical.</p>

	<p>Proposal: pass as payload (as the size of the parameter could be too big, e.g. when requesting storage commitment for an entire MR study).</p> <p>Decision (2022-09-16 WG06): as proposed; agreed with justification.</p>
9	<p>Issue: Do we add the Referenced Study and Series UIDs (optionally) to the Instance UIDs of the request and response payload? It may be easier for the Origin Server to find the appropriate Instance UIDs having these at hand.</p> <p>Proposal: Yes, the user agent may add the Referenced Study and Series UIDs, and the Origin Server is required to accept them.</p> <p>Repercussion: when the Instance UID is not part of the Study or Series that is provided in the request, a new kind of error is possible.</p> <p>Decision (2022-09-16 WG06): as proposed; understood the justification. Applicable failure code will be (the already existing) 0112H – No such object instance.</p>
10	<p>Issue: It should be easy to write a wrapper from classic DICOM to this DICOMweb variant of Storage Commit. This should be the case for both the SCP and the SCU.</p> <p>Proposal: The wrappers as described (from classic DICOM to DICOMweb) are relatively easy to create.</p> <p>Note that the proposals for Issue #5 and Issue #7 remove the Referenced SOP Class UIDs, and when there are current implementations that use these UIDs, reverse wrappers (unwrappers?!) cannot be created. This is, however, not perceived as an issue.</p> <p>Decision (2022-09-16 WG06): make it backwards compatible, so that it is easy to write a wrapper in two directions. So keep the information model as is.</p>
11	<p>Issue: Should the Commit service also be in PS3.18, section 9. URI Service?</p> <p>Proposal: no, as store is not part of this service.</p> <p>Decision (2022-09-19 WG06): As proposed.</p>
12	<p>Issue [WG27]: Is it possible to make the Studies and Series references mandatory?</p> <p>Proposal: Given backwards compatibility with DIMSE, this is not possible, but we can define a structure that allows for mandatory attributes when the user agent wants to. See also issue 14.</p> <p>Decision: As proposed.</p>
13	<p>Issue: What should be the way to pass the transaction UID to the result check? There are two options: as an HTTP parameter or as Payload.</p> <p>Proposal: pass as Payload, as a) this is similar to the request, and b) the UIDs will not show up in HTTP logging, preventing possible security issues.</p> <p>Decision: As proposed.</p>
14	<p>Issue (follow-up of issue 9): How are the Study and Series UID incorporated in the data structure as specified in the Storage Commitment Request – Action Information and as specified in the Storage Commitment Result – Event Information? There are three alternatives:</p> <ul style="list-style-type: none"> • <i>Straightforward</i> – Add the Referenced Study Instance UID and the Referenced Series Instance UID as optional items in the Referenced SOP Sequence (and in the Failed SOP Sequence). This is quite easy to specify and understand but will create a lot of duplicate data. • <i>Innovative</i> – Create an entirely new data structure, in which there is a Referenced Study Sequence. This holds the Referenced Study Instance UID, and the Referenced Series Sequence. This series sequence holds the Referenced Series Instance UID, and the Referenced SOP Sequence. The latter is defined as in DIMSE. Similarly, this nested approach could be taken for the Failed SOP Sequence too. The advantage of this would be the lack of duplication (imagine a 64K slide fMRI scan), really appreciated in

	<p>RESTful/cloud worlds; the disadvantage is that it is not backwards compatible, so in conflict with the conclusion of issue 10 (be able to create wrappers).</p> <ul style="list-style-type: none"><li data-bbox="261 262 1367 514">• <i>Hybrid</i> – Extend the existing data structure with an optional structure like in the second bullet. In this case, there should be the condition that it is not possible that both the (original) Referenced SOP Sequence and the (new) Referenced Study Sequence are non-empty. In that case, this new structure can be viewed as an either-or of the original structure and a nested structure. The advantage of this approach is that it is very flexible, both backwards compatible and RESTful/cloud-proof. The disadvantage is that there is a structure with duplication in the definition, that has conditions that need to be checked run-time (to prevent duplication in the instances). <p>Proposal: Go for the third option. The first approach will duplicate a lot (think about fMRI studies with 64K SOP Instances and requesting for storage commit for a day's production of these in one go). The second prevents bi-directionally proxyability.</p> <p>Decision: As proposed.</p>
--	--

Scope and Field of Application

This supplement defines the means to perform storage commitment in DICOMweb. This is an extension to the existing DICOMweb services, mimicking the storage commitment service that is already available using DIMSE. The storage commitment service is for safekeeping data.

This supplement provides updates to PS3.18 (Web Services) mainly; consequently, some updates to PS3.2 (Conformance), PS3.6 (Data Dictionary), and PS3.15 (Security and System Management Profiles) are required too.

Changes to NEMA Standards Publications PS 3.18

Add a new reference to section 2.

2 Normative References

...

2.3 Other References

...

[ARRp] *Asynchronous Request-Reply pattern*, <https://learn.microsoft.com/en-us/azure/architecture/patterns/async-request-reply>.

...

Update existing sections 10.1-10.3 as given below.

10 Studies Service and Resources

10.1 Overview

The Studies Resource enables a user agent to store, retrieve, update, and search an origin server for DICOM Studies, Series, and Instances, along with their /metadata, /rendered, and /thumbnail variants; as well as Frames and Bulkdata. **In addition, the Studies Service enables a user agent to arrange safe-keeping Instances at an origin server.**

The Retrieve transaction of this Service is also known as WADO-RS. The Store transaction of this Service is also known as STOW-RS. The Search transaction of this Service is also known as QIDO-RS. See Section 10.3.

10.1.1 Resource Descriptions

The Studies Service manages a collection of DICOM Study resources. Each Study is organized in a hierarchy of sub-resources that correspond to the DICOM Information Model. See Section 7 “DICOM Model of the Real World” in PS3.3.

There are ~~three~~**four** top level resources:

/studies	references all Studies managed by the service.
/series	references all Series managed by the service.
/instances	references all Instances managed by the service.
/commit	arranges storage commitment of Instances managed by the service.

The following URI Template variables are used in resource definitions in this Section.

{study}	the Study Instance UID of a Study managed by the Studies Service.
---------	---

{series}	the Series Instance UID of a Series contained within a Study resource.
{instance}	the SOP Instance UID of an Instance contained within a Series resource.
{frames}	a comma-separated list of frame numbers, in ascending order, contained within an Instance.
{bulkdataURI}	an opaque URI that references a Bulkdata Value.

The Studies Service defines the following resources:

Resource	Description
Studies Service	The Base URI of the Studies Service.
All Studies	The All Studies resource references the entire collection of Studies contained in the Studies Service.
Study	The Study resource references a single Study.
Study Metadata	The Study Metadata resource references the Metadata of a Study.
Study Bulkdata	The Study Bulkdata resource references the Bulkdata of a Study.
Study Pixel Data	The Study Pixel Data resource references the Pixel Data of a Study.
Rendered Study	The Rendered Study resource references a Study to be rendered.
Study Thumbnail	The Study Thumbnail resource references a thumbnail image of a Study.
Study's Series	The Study's Series resource references the collection of all Series contained in a Study.
Study's Instances	The Study's Instances resource references the collection of all Instances in a single Study.
All Series	The All Series resource references the collection of all Series in all Studies contained in the Studies Service.
Series	The Series resource references a single Series.
Series Metadata	The Series Metadata resource contains the Metadata of a Series in a Study.
Series Bulkdata	The Series Bulkdata resource references the Bulkdata of a Series.
Series Pixel Data	The Series Pixel Data resource references the Pixel Data of a Series.
Rendered Series	The Rendered Series resource references a Series to be rendered.
Series Thumbnail	The Series Thumbnail resource references a thumbnail image of a Series.
Series' Instances	The Series' Instances resource references the collection of all Instances in a single Series.
All Instances	The All Instances resource references the collection of all Instances in all Series in all Studies contained in the Studies Service.
Instance	The Instance resource references a single Instance.
Instance Metadata	The Instance Metadata resource contains the Metadata of an Instance.
Instance Bulkdata	The Instance Bulkdata resource references the Bulkdata of a Instance.
Instance Pixel Data	The Instance Pixel Data resource references the Pixel Data of a Instance.
Rendered Instance	The Rendered Instance resource references an Instance to be rendered.
Instance Thumbnail	The Instance Thumbnail resource references a thumbnail image of an Instance.
Frames	The Frames resource references an ordered collection of frames in a single multi-frame Instance.
Rendered Frames	The Rendered Frames resource references an ordered collection of frames of a single multi-frame Instance, to be rendered.
Frame Thumbnail	The Frame Thumbnail resource references a thumbnail image for frames within an Instance.
Bulkdata	The Bulkdata resource contains a Bulkdata Value.
Commit	The commit resource arranges storage commitment of SOP instances.

Note

There is no Frame Bulkdata or Frame Pixel Data resource because they would be equivalent to the Frames resource.

10.1.2 Common Query Parameters

The origin server shall support Query Parameters as required in Table 10.1.2-1.

The user agent shall supply in the request Query Parameters as required in Table 10.1.2-1.

Table 10.1.2-1. Common Query Parameters

Name	Value	Usage		Section
		User Agent	Origin Server	
Accept	media-type	O	M	Section 8.3.3.1
Accept-Charset	charset	O	M	Section 8.3.3.2

10.1.3 Common Media Types

The origin server media type requirements are defined in each Transaction of this Service.

The origin server shall support the Transfer Syntax and Character Set media type parameters. See Section 8.7.3.5.2 and Section 8.7.3.5.3.

10.2 Conformance

An origin server claiming conformance to the Retrieve Transaction of the Studies Service:

- shall support the Retrieve Capabilities Transaction (see Section 8.9.1);
- shall support the Retrieve Transaction for all mandatory resources in Table 10.3-2.

An origin server claiming conformance to the Store Transaction of the Studies Service:

- shall support the Retrieve Capabilities Transaction (see Section 8.9.1);
- shall support the Store Transaction for all mandatory resources in Table 10.3-2.

An origin server claiming conformance to the Search Transaction of the Studies Service:

- shall support the Retrieve Capabilities Transaction (see Section 8.9.1);
- shall support the Search Transaction for all mandatory resources in Table 10.3-2.

An origin server claiming conformance to the Commit Transaction of the Studies Service:

- **shall support the Retrieve Capabilities Transaction (see Section 8.9.1);**
- **shall support the Commit Transaction for all mandatory resources in Table 10.3-2.**

The user agent may support any of the transactions for any of the corresponding resources in Table 10.3-2.

10.3 Transactions Overview

The Studies Service consists of the following transactions:

Table 10.3-1. Studies Service Transactions

	Method	Payload	Description
--	--------	---------	-------------

Transaction Name		Request	Success Response	
Retrieve	GET	N/A	Instance(s), Metadata, Renderings, Pixel Data, or Bulkdata	Retrieve one or more representations of DICOM Resources.
Store	POST	Instance(s)	Store Instances Response Module	Stores one or more representations of DICOM Resources, contained in the request payload, in the location referenced by the Target Resource.
Search	GET	N/A	Result(s)	Searches the Target Resource for DICOM objects that match the search parameters and returns a list of matches in an Acceptable Media Type.
Commit	POST	<u>Transaction UID and Instance UIDs</u>	<u>OK: Result</u>	<u>Per referenced Instance, commits or fails to commit to safekeeping. Returns commitment status per referenced Instance.</u>
			<u>Accepted: N/A</u>	<u>Working to commit to safekeeping the referenced Instances.</u>
	GET	<u>Transaction UID</u>	<u>OK: Result</u>	<u>Per referenced Instance, commits or fails to commit to safekeeping. Returns commitment status per referenced Instance.</u>
			<u>Accepted: N/A</u>	<u>Still working to commit to safekeeping the referenced Instances.</u>

In Table 10.3-2, the Target Resources permitted for each transaction are marked with M if support is mandatory for the origin server and O if it is optional. A blank cell indicates that the resource is not allowed in the transaction.

Table 10.3-2. Resources by Transaction

Resource	Retrieve	Store	Search	<u>Commit</u>
Studies Service				
All Studies		M	M	
Study	M	M		
Study Metadata	M			
Study Bulkdata	O			
Study Pixel Data	O			
Rendered Study	M			
Study Thumbnail	O			
Study's Series			M	
Study's Instances			M	
All Series			M	
Series	M			
Series Metadata	M			
Series Bulkdata	O			
Series Pixel Data	O			
Series' Instances			M	
Rendered Series	M			
Series Thumbnail	O			

All Instances			M	
Instance	M			
Instance Metadata	M			
Instance Bulkdata	O			
Instance Pixel Data	O			
Rendered Instance	M			
Instance Thumbnail	O			
Frames	M			
Rendered Frames	M			
Frame Thumbnail	O			
Bulkdata	M	M		
Commit				M

Add new section 10.x Commit Transaction

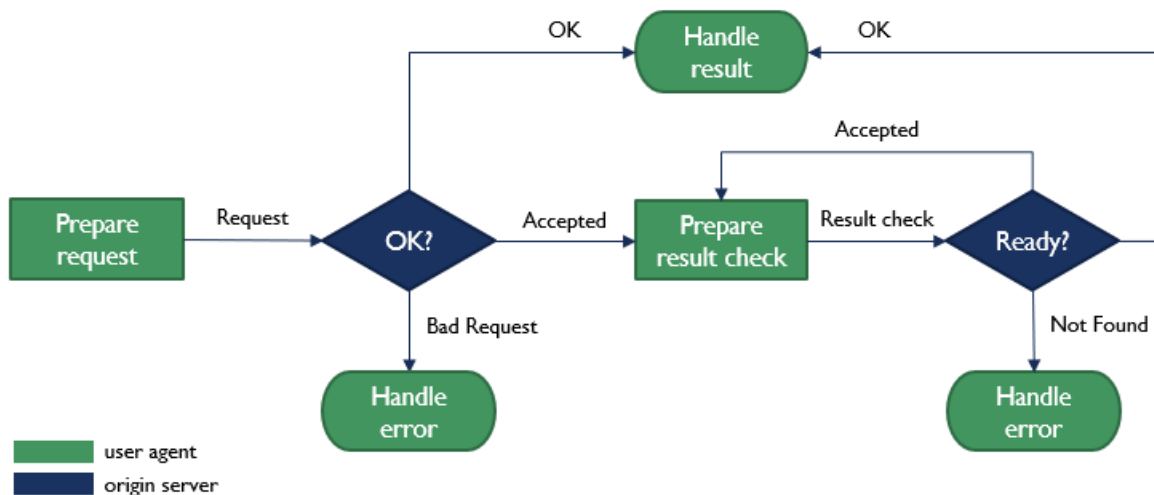
10.x Commit Transaction

This transaction uses the POST and GET methods to request storage commitment of Instances on the origin server. It corresponds to the DIMSE Storage Commitment Service Class as defined in Annex J of PS3.4, and has the same semantics.

As committing to storage of Instances is often a long-running operation on the origin server, this transaction may be split into two parts, to the discretion of the origin server: 1) requesting the commitment, and – when the origin server cannot give the result yet – 2) checking for the result, in line with the asynchronous request-reply pattern [ARRp].

Note A PACS may wait with a response to the commit request it receives, for instance until the VNA that it uses for long term storage has given commitment for the referenced Instances.

The diagram below shows the possible scenarios of the commit transaction. This starts when the user agent sends a commit Request to the origin server. This requests the origin server's commitment to safe-keep a set SOP Instances, specified by their respective UID.



In case the origin server responds with OK, it behaves synchronously, and returns, for each instance, whether it commits to safekeeping that instance, or not; the user agent can handle this result appropriately, for example by deleting the local copies of the instances that now are safely kept by the origin server.

In case the origin server responds to the commit Request with Accepted, it behaves asynchronously, and is effectively saying that it is working on the request. In this case, the user agent needs to perform a Result Check. When this check is performed, the origin server may response with OK, and will provide the same kind of result as in the synchronous case, which can be handled in the same way by the user agent. The origin server may also response to the Result Check with Accepted, which will trigger the user agent to perform a Result Check yet again. This process continues until the origin server responds with OK, finalizing the transaction.

For both the Request and the Result Check it is also possible that the origin server returns with an error, and this also needs to be handled appropriately by the user agent.

Note In line with section J.1.1 of PS3.4, the possession of a link to access pixel data shall not be sufficient for the origin server to commit to storage. A copy of the entire pixel data is required.

10.x.1 Request

The request shall have the following syntax:

```
POST SP "/" {/resource} SP version CRLF
Accept: 1#media-type CRLF
*(header-field CRLF)
CRLF
Payload
```

10.x.1.1 Target Resources

Table 10.x.1-1 defines the resources used to request storage commitment.

Table 10.x.1-1. Commit Storage Resources

Resource	URI Template
Commit	/commit

10.x.1.2 Query Parameters

The Commit transaction has no Query Parameters.

10.x.1.3 Request Header Fields

The origin server shall support Request Header Fields as required in Table 10.x.1-2.

The user agent shall supply Request Header Fields as required in Table 10.x.1-2.

Table 10.x.1-2. Request Header Fields

Name	values	Usage		Description
		User Agent	Origin Server	
Accept	media-type	M	M	The Acceptable Media Types of the response payload.

See also Section 8.4.

10.x.1.4 Request Payload

The request payload shall be present and shall contain one representation consistent with the Content-Type header field. The representation shall conform to Media Types described in Section 8.7.3 DICOM Media Type Sets. The payload shall conform to Section 8.6 Payloads.

The request payload shall contain the Referenced SOP Instance UIDs that the user agent requests the origin server to commit storage for.

The user agent shall supply the attributes mentioned in Table 10.x.1-3.

Table 10.x.1-3. Commit Request Payload

Attribute Name	Tag	Type	Attribute Description
Transaction UID	(0008,1195)	1	Uniquely identifies this Storage Commitment transaction.
Referenced SOP Sequence	(0008,1199)	1C	A Sequence of items where each item references a single SOP Instance for which storage commitment is requested. Required only if the Referenced Study Instance Sequence is absent (mutually exclusive).
<i>>Table 10-11 "SOP Instance Reference Macro Attributes" in PS3.3</i>			
Referenced Study Instance Sequence	(0008,xxx1)	1C	A Sequence of items where each item references a single Study for which (partial) storage commitment is requested. Required only if the Referenced SOP Sequence is absent (mutually exclusive).
>Referenced Study Instance UID	(0008,xxx2)	1	Uniquely identifies the referenced Study.
>Referenced Series Instance Sequence	(0008,xxx3)	1	A Sequence of items where each item references a single Series for which (partial) storage commitment is requested.
>>Referenced Series Instance UID	(0008,xxx4)	1	Uniquely identifies the referenced Series.
>>>Referenced SOP Class Sequence	(0008,xxx5)	1	A Sequence of items where each item references a single SOP Class for which (partial) storage commitment is requested.
>>>>Referenced SOP Class UID	(0008,1150)	1	Uniquely identifies the referenced SOP Class.
>>>>Referenced SOP Instance Sequence	(0008,xxx6)	1	A Sequence of items where each item references a single SOP Instance for which storage commitment is requested.
>>>>>Referenced SOP Instance UID	(0008,1155)	1	Uniquely identifies the referenced SOP Instance.

10.x.2 Behavior

The origin server shall commit or reject safekeeping the SOP Instances referenced by the Referenced SOP Instance UIDs. It will make clear what Instances have been committed to safekeep, and what Instances were rejected for safekeeping.

10.x.3 Response

The response shall have the following syntax:

```
version SP status-code SP reason-phrase CRLF
[retry-after CRLF]
CRLF
[Payload]
```

10.x.3.1 Status Codes

Table 10.x.3-1 shows some common status codes corresponding to this transaction. See also Section 8.5 for additional status codes.

Table 10.x.3-1. Status Code Meaning

Status	Code	Meaning
Success	200 (OK)	The origin server finished processing the request; the result is found in the Payload, which shall be present, see section 10.x.3.3; the retry-after parameter shall be absent.
	202 (Accepted)	The origin server has not finished processing the request yet; the retry-after parameter may be present; there shall not be a Payload.
Failure	400 (Bad Request)	The origin server cannot process the request because of errors in the request headers or parameters. Neither the retry-after parameter nor the Payload shall be present.

In case of OK, the Payload describes in detail what SOP Instances have been committed for safekeeping, and what instances have not been committed for safekeeping by the origin server.

10.x.3.2 Response Header Fields

The origin server shall support header fields as required in Table 10.x.3-2.

Table 10.x.3-2. Response Header Fields

Name	Value	Origin Server Usage	Description
Content-Type	media-type	M	The media type of the response payload, if present.
Content-Encoding	encoding	O	Shall be present if the response payload has a content encoding. See Section 8.4.3.
Content-Length	uint	O	Shall be present if the response payload does not have a content encoding. See Section 8.4.3.
Retry-After	uint	O	The number of seconds the user agent is requested to wait until (a next) result check.

All success responses shall also contain the Content Representation (see Section 8.4.2) and Payload header fields (see Section 8.4.3) with appropriate values.

It is recommended that the text returned in the Warning header field (see [RFC7234] Section 5.5) contain a DICOM Status Code (see PS3.4 and Annex C “Status Type Encoding (Normative)” in PS3.7) and descriptive reason. For example:

```
Warning: A700 <service>: Out of memory
```

10.x.3.3 Response Payload

The response payload shall contain attributes in accordance with Table 10.x.3-3.

Note The response payload shall only be present in case of the success status code 200 (OK).

Table 10.x.3-3. Response Payload

Attribute Name	Tag	Type	Attribute Description
Transaction UID	(0008,1195)	1	Uniquely identifies the Storage Commitment transaction.
Referenced SOP Sequence	(0008,1199)	1C	A Sequence of items where each item references a single SOP Instance for which storage has been committed. Required if the request payload contained the Referenced SOP Sequence, and there is at least one SOP Instance in that supplied sequence for which storage has been committed; shall not be present otherwise.
<i>>Table 10-11 "SOP Instance Reference Macro Attributes" in PS3.3</i>			
Failed SOP Sequence	(0008,1198)	1C	A sequence of items where each item references a single SOP Instance for which storage has not been committed. Required if the request payload contained the Referenced SOP Sequence, and there is at least one SOP Instance in that supplied sequence for which storage has not been committed; shall not be present otherwise.
<i>>Table 10-11 "SOP Instance Reference Macro Attributes" in PS3.3</i>			
>Failure Reason	(0008,1197)	1	The reason that storage commitment could not be provided for this SOP Instance. See PS3.3, section C.14.1.1 for possible values.

Referenced Study Instance Sequence	(0008,xxx1)	1C	A Sequence of items where each item references a single Study for which (partial) storage has been committed. Required if the request payload contained the Referenced Study Instance Sequence, and there is at least one SOP Instance for which storage has been committed; shall not be present otherwise.
>Referenced Study Instance UID	(0008,xxx2)	1	Uniquely identifies the referenced Study.
>Referenced Series Instance Sequence	(0008,xxx3)	1	A Sequence of items where each item references a single Series for which (partial) storage has been committed.
>>Referenced Series Instance UID	(0008,xxx4)	1	Uniquely identifies the referenced Series.
>>Referenced SOP Class Sequence	(0008,xxx5)	1	A Sequence of items where each item references a single SOP Class for which (partial) storage has been committed.
>>>Referenced SOP Class UID	(0008,1150)	1	Uniquely identifies the referenced SOP Class.
>>>Referenced SOP Instance Sequence	(0008,xxx6)	1	A Sequence of items where each item references a single SOP Instance for which storage has been committed.
>>>>Referenced SOP Instance UID	(0008,1155)	1	Uniquely identifies the referenced SOP Instance.

Failed Study Instance Sequence	(0008,xxx7)	1C	A Sequence of items where each item references a single Study for which (partial) storage commitment has been rejected. Required if the request payload contained the Referenced Study Instance Sequence, and there is at least one SOP Instance in that supplied sequence for which storage has not been committed; shall not be present otherwise.
>Referenced Study Instance UID	(0008,xxx2)	1	Uniquely identifies the referenced Study.
>Referenced Series Instance Sequence	(0008,xxx3)	1	A Sequence of items where each item references a single Series for which (partial) storage commitment has been rejected.
>>Referenced Series Instance UID	(0008,xxx4)	1	Uniquely identifies the referenced Series.
>>Referenced SOP Class Sequence	(0008,xxx5)	1	A Sequence of items where each item references a single SOP Class for which (partial) storage commitment has been rejected.
>>>Referenced SOP Class UID	(0008,1150)	1	Uniquely identifies the referenced SOP Class.
>>>Referenced SOP Instance Sequence	(0008,xxx6)	1	A Sequence of items where each item references a single Instance for which storage commitment has been rejected.
>>>>Referenced SOP Instance UID	(0008,1155)	1	Uniquely identifies the referenced SOP Instance.
>>>>Failure Reason	(0008,1197)	1	The reason that storage commitment could not be provided for this SOP Instance.

10.x.4 Result Check

The result check shall have the following syntax:

```
GET SP "/" {/resource} SP version CRLF
Accept: 1#media-type CRLF
*(header-field CRLF)
CRLF
Payload
```

Note The result needs only to be checked in case the origin server has responded that it has accepted but not yet completed the commit request (returning status code 202).

10.x.4.1 Target Resources

Table 10.x.4-1 defines the resources used to check the result of requesting storage commitment.

Table 10.x.4-1. Result Check Resources

Resource	URI Template
Commit	/commit

10.x.4.2 Query Parameters

The result check transaction has no Query Parameters.

10.x.4.3 Result Check Header Fields

The origin server shall support Result Check Header Fields as required in Table 10.x.4-2.

The user agent shall supply Result Check Header Fields as required in Table 10.x.4-2.

Table 10.x.4-2. Result Check Header Fields

Name	values	Usage		Description
		User Agent	Origin Server	
Accept	media-type	M	M	The Acceptable Media Types of the response payload.

See also Section 8.4.

Note The presence and values of the result check header fields should be the same as those of the request header fields.

10.x.4.4 Result Check Payload

The request payload shall be present and shall contain one representation consistent with the Content-Type header field. The representation shall conform to Media Types described in Section 8.7.3 DICOM Media Type Sets. The payload shall conform to Section 8.6 Payloads.

The user agent shall supply the attributes mentioned in Table 10.x.4-3.

Table 10.x.4-3. Result Check Payload

Attribute Name	Tag	Type	Condition
Transaction UID	(0008,1195)	M	

10.x.5 Result

The result shall have the following syntax:

```
version SP status-code SP reason-phrase CRLF
[retry after CRLF]
CRLF
[Payload]
```

10.x.5.1 Status Codes

Table 10.x.5-1 shows some common status codes corresponding to this transaction. See also Section 8.5 for additional status codes.

Table 10.x.5-1. Status Code Meaning

Status	Code	Meaning
Success	200 (OK)	The origin server finished processing the request; the result is found in the Payload, which shall be present, see section 10.x.5.3; the retry-after parameter shall be absent.
	202 (Accepted)	The origin server has not finished processing the request yet; the retry-after parameter may be present; there shall not be a Payload.

Failure	404 (Not Found)	The origin server could not find the request with the supplied Transaction UID; both the retry-after parameter and the Payload shall be absent.
---------	-----------------	---

In case of OK, the Payload describes in detail what SOP Instances have been committed for safekeeping, and what instances have not been committed for safekeeping by the origin server.

Note When the 'Not Found' status code is returned, this means that either a wrong Transaction UID has been supplied, or that the status code of the response has been 200 (OK), or that the user agent waited too long with the result check. In the latter two cases, the origin server may have deleted the applicable results.

Note When the 'Not Found' status code is returned, the user agent may simply retry with a new storage commit request.

10.x.5.2 Result Header Fields

See Section 10.x.3.2.

10.x.5.3 Response Payload

See Section 10.x.3.3.

10.x.6 Media Types

The origin server shall support the media types specified as default or required in Table 10.x.6-1.

Table 10.x.6-1. Commit Response Payload

Media Type	Usage	Section
application/dicom+json	Default	Section 8.7.3.2
application/dicom+xml	Required	Section 8.7.3.2
multipart/related; type="application/dicom+xml"	Required	Section 8.7.3.2
multipart/related; type="application/dicom+xml"	Required	Section 8.7.3.2

10.x.7 Conformance Statement

An implementation claiming conformance to the Commit transaction shall support the resources and media types as specified in Section 10.x.

An implementation shall declare in its Conformance Statement whether it plays the role of origin server or user agent, or both.

Implementation specific warning and error codes shall be included in the Conformance Statement.

When an implementation claims to play the role of origin server, it shall specify in its Conformance Statement how long – in number of hours – the result of a request will be kept. This is either the number of hours after the result has become available or the number of hours after the first (unsuccessful) result check, whatever comes last.

Note The origin server may delete the local copy of the result after it has been passed to the user agent.

Add new examples for the Commit Transaction

B Examples (Informative)

...

B.x1 Request Commit for Multiple Instances with JSON

Sending the POST request for multiple instances:

```
POST /radiology/commit HTTP/1.1
Host: www.hospital-stmarco
Content-Type: application/dicom+json
...
{
  "00081195": {
    "vr": "UI",
    "Value": [
      "1.1.99999.20220901"
    ]
  },
  "00081199": {
    "vr": "SQ",
    "Value": [
      {
        "00081150": {
          "vr": "UI",
          "Value": [
            "1.2.840.10008.5.1.4.1.1.2"
          ]
        },
        "00081155": {
          "vr": "UI",
          "Value": [
            "1.3.12.2.1107.5.99.3.30000012031310075961300000059"
          ]
        }
      ]
    ]
  },
  "00081150": {
    "vr": "UI",
    "Value": [
      "1.2.840.10008.5.1.4.1.1.2"
    ]
  },
  "00081155": {
    "vr": "UI",
    "Value": [
      "1.3.12.2.1107.5.99.3.30000012031310075961300000060"
    ]
  }
},
...
]
```

Response to the request (this is the case where the server notifies the user agent should wait before making a follow-up request, the direct response case is mentioned below)

```
HTTP/1.1 202 Accepted
Retry-After: 300
...
```


Afterwards GET is called to get the status of the request:

```
GET /radiology/commit HTTP/1.1
Host: www.hospital-stmarco
Content-Type: application/dicom+json
...
{
  "00081195": {
    "vr": "UI",
    "Value": [
      "1.1.99999.20220901"
    ]
  }
}
```

Response to the check status (in case that the server initially responds to the POST request with the HTTP response status code 200 and no Retry-After HTTP header the following content we would get directly after initial POST request):

```
HTTP/1.1 200 OK
Content-Length: 1191
Content-Type: application/dicom+json; charset=utf-8
...
{
  "00081195": {
    "vr": "UI",
    "Value": [
      "1.1.99999.20220901"
    ]
  },
  "00081199": {
    "vr": "SQ",
    "Value": [
      {
        "00081150": {
          "vr": "UI",
          "Value": [
            "1.2.840.10008.5.1.4.1.1.2"
          ]
        },
        "00081155": {
          "vr": "UI",
          "Value": [
            "1.3.12.2.1107.5.99.3.30000012031310075961300000059"
          ]
        }
      }
    ]
  },
  "00081198": {
    "vr": "SQ",
    "Value": [
      {
        "00081150": {
          "vr": "UI",
          "Value": [
            "1.2.840.10008.5.1.4.1.1.2"
          ]
        },
        "00081155": {
          "vr": "UI",
          "Value": [
            "1.3.12.2.1107.5.99.3.30000012031310075961300000060"
          ]
        }
      }
    ]
  },
  "00081197": {
    "vr": "US",
    "Value": [ 274 ]
  }
},
```

```
    ]  
}
```

B.x2 Request Commit for Multiple Instances with XML and Referenced Study and Series Instance UIDs

```
POST /radiology/commit HTTP/1.1  
Host: www.hospital-stmarco  
Content-Type: application/dicom+xml  
...  
<?xml version="1.0" encoding="UTF-8"?>  
<NativeDicomModel>  
  <DicomAttribute Tag="00081195" VR="UI" keyword="Transaction UID">  
    <Value number="1">1.2.99999.20220902</value>  
  </DicomAttribute>  
  <DicomAttribute Tag="0008XXX1" VR="SQ" keyword="Referenced Study Instance Sequence">  
    <Item number="1">  
      <DicomAttribute Tag="0008XXX2" VR="UI" Keyword="Referenced Study Instance UID">  
        <Value number="1">1.2.250.1.59.40211.12345678.678910</value>  
      </DicomAttribute>  
      <DicomAttribute Tag="0008XXX3" VR="SQ" keyword="Referenced Series Instance Sequence">  
        <Item number="1">  
          <DicomAttribute Tag="0008XXX4" VR="UI" Keyword="Referenced Series Instance UID">  
            <Value number="1">1.2.250.1.59.40211.789001276.14556172.67789</value>  
          </DicomAttribute>  
          <DicomAttribute Tag="0008XXX5" VR="SQ" keyword="Referenced SOP Class Sequence">  
            <Item number="1">  
              <DicomAttribute Tag="00081150" VR="UI" Keyword="Referenced SOP Class UID">  
                <Value number="1">1.2.840.10008.5.1.4.1.1.4</value>  
              </DicomAttribute>  
              <DicomAttribute Tag="0008XXX6" VR="SQ" keyword="Referenced SOP Instance Sequence">  
                <Item number="1">  
                  <DicomAttribute Tag="00081155" VR="UI" Keyword="Referenced SOP Instance UID">  
                    <Value number="1">1.2.250.1.59.40211.2678810.87991027.899772.1</value>  
                  </DicomAttribute>  
                </DicomAttribute>  
              </DicomAttribute>  
            </Item number="1">  
          </DicomAttribute>  
        </Item number="1">  
      </DicomAttribute>  
    </DicomAttribute>  
  </NativeDicomModel>  
  
HTTP/1.1 202 Accepted  
Retry-After: 120  
...  
  
GET /radiology/commit HTTP/1.1  
Host: www.hospital-stmarco  
Content-Type: application/dicom+xml  
...  
<?xml version="1.0" encoding="UTF-8"?>  
<NativeDicomModel>  
  <DicomAttribute Tag="00081195" VR="UI" keyword="Transaction UID">  
    <Value number="1">1.2.99999.20220902</value>  
  </DicomAttribute>  
</NativeDicomModel>  
  
HTTP/1.1 200 OK  
Content-Length: 826  
Content-Type: application/dicom+xml  
...  
<?xml version="1.0" encoding="UTF-8"?>  
<NativeDicomModel>  
  <DicomAttribute Tag="00081195" VR="UI" keyword="Transaction UID">  
    <Value number="1">1.2.99999.20220902</value>
```

```
</DicomAttribute>
<DicomAttribute Tag="0008XXX1" VR="SQ" keyword="Referenced Study Instance Sequence">
  <Item number="1">
    <DicomAttribute Tag="0008XXX2" VR="UI" Keyword="Referenced Study Instance UID">
      <Value number="1">1.2.250.1.59.40211.12345678.678910</value>
    </DicomAttribute>
    <DicomAttribute Tag="0008XXX3" VR="SQ" keyword="Referenced Series Instance Sequence">
      <Item number="1">
        <DicomAttribute Tag="0008XXX4" VR="UI" Keyword="Referenced Series Instance UID">
          <Value number="1">1.2.250.1.59.40211.789001276.14556172.67789</value>
        </DicomAttribute>
        <DicomAttribute Tag="0008XXX5" VR="SQ" keyword="Referenced SOP Class Sequence">
          <Item number="1">
            <DicomAttribute Tag="00081150" VR="UI" Keyword="Referenced SOP Class UID">
              <Value number="1">1.2.840.10008.5.1.4.1.1.4</value>
            </DicomAttribute>
            <DicomAttribute Tag="0008XXX6" VR="SQ" keyword="Referenced SOP Instance Sequence">
              <Item number="1">
                <DicomAttribute Tag="00081155" VR="UI" Keyword="Referenced SOP Instance UID">
                  <Value number="1">1.2.250.1.59.40211.2678810.87991027.899772.1</value>
                </DicomAttribute>
              </Item number="1">
            </DicomAttribute>
          </Item number="1">
        </DicomAttribute>
      </Item number="1">
    </DicomAttribute>
  </Item number="1">
</DicomAttribute>
...
</DicomAttribute>
...
</DicomAttribute>
...
</DicomAttribute>
</NativeDicomModel>
```

B.x3 Request Commit with HTTP Multipart Request for Multiple Studies

```
POST /radiology/commit HTTP/1.1
Host: www.hospital-stmarco
Content-Type: multipart/related; type="application/dicom+xml"; boundary=MESSAGEBOUNDARY
...
--MESSAGEBOUNDARY
<?xml version="1.0" encoding="UTF-8"?>
<NativeDicomModel>
  <DicomAttribute Tag="00081195" VR="UI" keyword="Transaction UID">
    <Value number="1">1.2.99999.20220903</value>
  </DicomAttribute>
  <DicomAttribute Tag="0008XXX1" VR="SQ" keyword="Referenced Study Instance Sequence">
    <Item number="1">
      <DicomAttribute Tag="0008XXX2" VR="UI" Keyword="Referenced Study Instance UID">
        <Value number="1">1.2.250.1.59.40211.12345678.678910</value>
      </DicomAttribute>
      <DicomAttribute Tag="0008XXX3" VR="SQ" keyword="Referenced Series Instance Sequence">
        <Item number="1">
          <DicomAttribute Tag="0008XXX4" VR="UI" Keyword="Referenced Series Instance UID">
            <Value number="1">1.2.250.1.59.40211.789001276.14556172.67789</value>
          </DicomAttribute>
          <DicomAttribute Tag="0008XXX5" VR="SQ" keyword="Referenced SOP Class Sequence">
            <Item number="1">
              <DicomAttribute Tag="00081150" VR="UI" Keyword="Referenced SOP Class UID">
                <Value number="1">1.2.840.10008.5.1.4.1.1.4</value>
              </DicomAttribute>
              <DicomAttribute Tag="0008XXX6" VR="SQ" keyword="Referenced SOP Instance Sequence">
                <Item number="1">
                  <DicomAttribute Tag="00081155" VR="UI" Keyword="Referenced SOP Instance UID">
                    <Value number="1">1.2.250.1.59.40211.2678810.87991027.899772.1</value>
                  </DicomAttribute>
                </Item number="1">
              </DicomAttribute>
            </Item number="1">
          </DicomAttribute>
        </Item number="1">
      </DicomAttribute>
    </Item number="1">
  </DicomAttribute>
  ...
  </DicomAttribute>
  ...
  </DicomAttribute>
  ...
  </DicomAttribute>
  ...
  </DicomAttribute>
```


	metadata	Retrieve Study Metadata	Section 10.4
	series	Search for Study Series	Section 10.6
	{SeriesInstance}	Retrieve Series	Section 10.4
	metadata	Retrieve Series Metadata	Section 10.4
	instances	Search for Study Series Instances	Section 10.4
	{SOPInstance}	Retrieve Instance	Section 10.4
	metadata	Retrieve Instance Metadata	Section 10.4
	frames	N/A	N/A
	{framelist}	Retrieve Frames	Section 10.4
	instances	Search for Study Instances	Section 10.6
	series	Search for Series	Section 10.6
	{SeriesInstance}	N/A	N/A
	{instances}	Search for Instances	Section 10.6
	instances	Search for Instances	Section 10.6
	{BulkDataReference}	Retrieve Bulkdata	Section 10.4
	commit	Commit	Section 10.x
Worklist (see Section 11.1.1)			
	workitems	Search for Workitem Create Workitem	Section 11.9 Section 11.4
	{Workitem}	Retrieve Workitem Update Workitem	Section 11.4 Section 11.6
	state	Change Workitem State	Section 11.7
	cancelrequest	Request Workitem Cancellation	Section 11.8
	subscribers	N/A	N/A
	{AETitle}	Subscribe Unsubscribe	Section 11.10 Section 11.11
	1.2.840.10008.5.1.4.34.5	N/A	N/A
	subscribers	N/A	N/A
	{AETitle}	Subscribe Unsubscribe	Section 11.10 Section 11.11
	suspend	Unsubscribe	Section 11.11
	1.2.840.10008.5.1.4.34.5.1	N/A	N/A
	subscribers	N/A	N/A
	{AETitle}	Subscribe Unsubscribe	Section 11.10 Section 11.11
	suspend	Suspend Worklist Subscription	Section 11.11
Non-Patient Instance (see Section 12.1.1)			
	color-palettes	N/A	N/A
	{uid}	Retrieve Store Search	Section 12.4 Section 12.5 Section 12.6

	defined-procedure-protocol	N/A	N/A
	{uid}	Retrieve Store Search	Section 12.4 Section 12.5 Section 12.6
	hanging-protocol	N/A	N/A
	{uid}	Retrieve Store Search	Section 12.4 Section 12.5 Section 12.6
	implant-templates	N/A	N/A
	{uid}	Retrieve Store Search	Section 12.4 Section 12.5 Section 12.6
	inventories	N/A	N/A
	{uid}	Retrieve Store Search	Section 12.4 Section 12.5 Section 12.6

Changes to NEMA Standards Publications PS 3.2

Pending the approval of Supplement 209, add the new transaction to table A.1-9 in section A.1.3.2.

A DICOM Conformance Statement Template (Normative)

...

A.1 Overview

...

A.1.3 DICOM Web Services

...

A.1.3.2 Study Service

Table 0-1 lists details on the support of the Study Service.

[Complete Table 0-1 to indicate support for the Study Web Service]

Table 0-1 Study Service

Service	Transaction	Resource	User Agent	Origin Server
Study Web Service	Retrieve Capabilities <i>Retrieve (WADO-RS)</i>			
		Study		
		Study Metadata		
		<i>Study Bulkdata</i>		
		<i>Study Pixel Data</i>		
		Rendered Study		
		<i>Study Thumbnail</i>		
		Series		
		Series Metadata		
		<i>Series Bulkdata</i>		
		<i>Series Pixel Data</i>		
		Rendered Series		
		<i>Series Thumbnail</i>		
		Instance		
		Instance Metadata		
		Instance Bulkdata		
		<i>Instance Pixel Data</i>		
		Rendered Instance		
		<i>Instance Thumbnail</i>		
	Frames			
	Rendered Frames			
	<i>Frame Thumbnail</i>			
	Bulkdata			
<i>Search (QIDO-RS)</i>	All Studies			
	Study			
	Study's Series			

		Study's Instances		
		All Series		
		Series		
		Series Instances		
		All Instances		
		Instance		
	<i>Store (STOW-RS)</i>	All Studies		
		Study		
		Bulkdata		
	<u>Commit</u>	<u>commit</u>		

Pending the approval of Supplement 209, add a new subsection on the Commit Transaction to section A.5.3.2 Studies Web Service

A DICOM Conformance Statement Template (Normative)

...

A.5 Service and Interoperability Description

...

A.5.3 Supported DICOM Web Services

...

A.5.3.2 Studies Web Service

...

A.5.3.2.x Commit Transaction

A.5.3.2.x.1 User Agent

The Commit Transaction user agent can request resources listed in Table A.5-xx1.

[List the supported resources for your Commit Transaction user agent. Remove the non-supported resources rows. Fill in information on your implementation in the Comments column when necessary.]

Table A.5-xx1: Resources Commit Transaction – User Agent

Resource	Comments
	See resource path in PS3.18 Table 10.x.1-1
<i>Commit</i>	

The Commit Transaction user agent supports Header Fields listed in Table A.5-xx2.

[List the supported Header Fields and their supported Values. Fill in information on your implementation in the “Comments” column when necessary.]

Table A.5-xx2: Header Fields for Commit Transaction – User Agent

Header Field	Supported Values	Comments
--------------	------------------	----------

Content-Type	application/dicom+json application/dicom+xml multipart/related; type="application/dicom+json" multipart/related; type="application/dicom+xml"	
Content-Length		<i>[If Content-Encoding is not present]</i>
Content-Encoding		<i>[If Content-Length is not present]</i>

A.5.3.2.x.2 Origin Server

The Commit Transaction origin server receives POST requests and possibly GET requests to enable storage commitment of the supplied SOP Instances.

The user agent specifies the Target Resource as part of the URI and specifies the UIDs of the SOP Instances as part of the data in the request body with a proper Content-Type (i.e. XML or JSON). The URI is composed by a Base URI: See Base URI for the origin server in Section A.6.3.2.x. The Commit Transaction origin server supports resources listed in Table A.5-xx3.

[Fill in information on your implementation in the Comments column when necessary.]

Table A.5-xx3: Resources Commit Transaction - Origin Server

Resource	Comments
	See resource path in PS3.18 Table 10.x.1-1
commit	

The Commit Transaction origin server supports Header Fields listed in Table A.5-xx4.

[List the supported Header Fields and their supported Values. Fill in information on your implementation in the "Comments" column when necessary.]

Table A.5-xx4: Header Fields for Commit Transaction - Origin Server

Header Field	Supported Values	Comments
Content-Type	application/dicom+json application/dicom+xml multipart/related; type="application/dicom+json" multipart/related; type="application/dicom+xml"	
Content-Length		<i>[If Content-Encoding is not present]</i>
Content-Encoding		<i>[If Content-Length is not present]</i>

Pending the approval of Supplement 209, add a new subsection on the Commit Transaction to section A.6.3.2 Studies Web Service.

A DICOM Conformance Statement Template (Normative)

...

A.6 Configuration

...

A.6.3 Configuration of DICOM Web Services

...

A.6.3.2 Studies Web Service

...

A.6.3.2.x Commit Transaction

Table A.6-xx5 lists configuration parameters for the Commit Transaction of the Studies Web Service:

[Remove the unsupported parameters from the local and remote configuration parameters.]

Table A.6-xx5: Commit Transaction Parameters

Local Configuration Parameters – Commit Transaction			
Parameter	Configurable	Default Value	Comments
	<<USER SERVICE FIXED>>	<i>[If there is no default, leave blank]</i>	<i>[Provide comments or Values/ranges if applicable]</i>
<i>Commit local Origin Server URL (Base URI)</i>	<i>FIXED</i>	<i>http://<host-name>:<port>/commit</i>	
<i>Port</i>	<i>SERVICE</i>	8081	
<i>Secured Commit local Origin Server URL (Base URI)</i>	<i>SERVICE</i>		
<i>Secured Port</i>	<i>SERVICE</i>		
<i><Specific Commit Transaction parameter></i>			
Remote Configuration Parameters - Commit Transaction			
<i>[Either document the number of supported remote hosts, e.g <Product> supports configuration of up to <X> remote hosts or state that there is no limitation other than the ones mandated by the operating system.]</i>			
Parameter	Configurable	Default Value	Comments
	<<USER SERVICE FIXED>>	<i>[If there is no default, leave blank]</i>	<i>[Provide comments or Values/ranges if applicable]</i>
<i>Commit remote Origin Server URL</i>	<i>USER</i>		
<i>Port</i>	<i>USER</i>		

Secured CommitRemote Origin Server URL	SERVICE		
Secured Port	SERVICE		
<Specific Commit Transaction parameter>			

Pending the approval of Supplement 209, add a new subsection on the Commit Transaction to section A.7.3.3.3 Studies Web Service.

A DICOM Conformance Statement Template (Normative)

A.7 Network and Media Communication Details

...

A.7.3 Status Codes

...

A.7.3.3 DICOM Web Services

...

A.7.3.3.3 Studies Web Service

...

A.7.3.3.3.7 Commit Transaction as Origin Server

Table A.7-xx6 lists the Status Codes that an origin server supports for the Retrieve Transaction of the Studies Web Service and the condition in which any of the listed Status Codes is sent:

[Describe below the condition in which the application sends the specific Status Codes in the Retrieve Transaction response as origin server.]

Table A.7-xx6: Status Codes of Origin Server for Commit Transaction

Status	Code	Condition
Success	200 (OK)	
	202 (Accepted)	
Failure	400 (Bad Request)	
	404 (Not Found)	

A.7.3.3.3.8 Commit Transaction as User Agent

Table A.7-xx7 lists the Status Codes that a user agent supports for the Retrieve Transaction of the Studies Web Service and the defines the application behavior, when encountering any of the listed Status Codes:

[Describe below the behavior of the application when it receives various Status Codes in the Retrieve Transaction response]

Table A.7-xx7: Status Codes of User Agent for Commit Transaction

Status	Code	Behavior
Success	200 (OK)	
	202 (Accepted)	
Failure	400 (Bad Request)	
	404 (Not Found)	
*	Any other code	

Changes to NEMA Standards Publications PS 3.6

Add the new attributes to table 6-1 of section 6.

6 Registry of DICOM Data Elements

...

Table 6-1. Registry of DICOM Data Elements

Tag	Name	Keyword	VR	VM	
...
(0008,xxx1)	<u>Referenced Study Instance Sequence</u>	<u>ReferencedStudyInstanceSe- quence</u>	<u>SQ</u>	<u>1</u>	
(0008,xxx2)	<u>Referenced Study Instance UID</u>	<u>ReferencedStudyInstanceUID</u>	<u>UI</u>	<u>1</u>	
(0008,xxx3)	<u>Referenced Series Instance Sequence</u>	<u>ReferencedSeriesInstanceSe- quence</u>	<u>SQ</u>	<u>1</u>	
(0008,xxx4)	<u>Referenced Series Instance UID</u>	<u>ReferencedSeriesInstanceUID</u>	<u>UI</u>	<u>1</u>	
(0008,xxx5)	<u>Referenced SOP Class Sequence</u>	<u>ReferencedSOPClassSequence</u>	<u>SQ</u>	<u>1</u>	
(0008,xxx6)	<u>Referenced SOP Instance Sequence</u>	<u>ReferencedSOPInstanceSe- quence</u>	<u>SQ</u>	<u>1</u>	
(0008,xxx7)	<u>Failed Study Instance Sequence</u>	<u>FailedStudyInstanceSequence</u>	<u>SQ</u>	<u>1</u>	
...					

Changes to NEMA Standards Publications PS 3.15

The new attributes need not be added to Table E.1-1 in section E.1.1, as they are never persisted.