Digital Imaging and Communications in Medicine (DICOM)

Supplement 234: DICOMweb Storage Commitment Service

Prepared by: Working Group 27

DICOM Standards Committee, Working Group 6

1300 N. 17th Street, Suite 900

Rosslyn, Virginia 22209 USA

Status: November 2023, Letter Ballot

Developed pursuant to DICOM Work Item 2022-04-A

Table of Contents

Document History

September 2022	Version 0.1	JM	Initial version, ready for First Read (FR) of WG06.
November 2022	Version 0.2	DK & JM	Updated with results from WG06's FR (September 2022): • Closed issues. • Synchronous and asynchronous behavior. Extended to be ready for discussion for Public Comment: • 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.
January 2023	Version 0.3	DK & JM	Updated with the results from WG06's November meeting: • Accepted all tracked textual changes made during the review of WG06. Copied these changes to applicable texts. • Made Storage Commitment a separate service, so moved it from the Studies Service and Resources (10.x) to a new top-level section (X). Added more textual changes.
March 2023	Version 0.4	DK & JM	Updated with the results from WG06's January meeting: • Accepted all tracked textual changes made during the review of WG06. Copied these changes to applicable texts. • Changed name of Supplement (added Service). • Changed the resource from `/commit` to `/commitment-requests`, to be in line with the other top-level resources. • Added {transactionUID} to the HTTP communication (and removed it from the payloads). Added a return code for handling the case of duplicate transaction UIDs. • Aligned the way of phrasing of this supplement with the existing parts of PS3.18. • Copied (and adapted, naming only) conformance requirements from the DIMSE section on the Storage Commitment Service (section J in PS3.4). Added more textual changes.
June 2023	Version 0.5	DK & JM	Updated with the results from WG06's March meeting: • Accepted all tracked textual changes, adapted text to selected tags, and removed resolved comments.

			• Moved request and response payloads as modules to Annex J. • Incorporated comments from WG31 (Sup209) on update of conformance statement template. Incorporated comments from WG06's June meeting.
August 2023	Draft Letter Ballot	JM	Incorporated Public Comments and comments made during line-by-line review in WG06. Elaborated on new items identified during the WG06 meeting.
November 2023	Letter Ballot	JM	Incorporated comments from continued line-by-line review in WG06. Elaborated on new items identified in WG06 meeting.

Open Issues

Closed Issues

1	**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)? **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." **Decision** (2022-09-19 WG-06): Agree with proposal. (2022-11-17 WG-06): Changed to making it a separate service, as there are two transactions.
2	**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 … 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. **Proposal**: the name of the service will be Commit Storage Service (as the Store service is not called Request to Store service either). **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. (2022-11-17 WG06): Since the service will get its own section (see 2nd decision in Closed Issue #1), its naming need not be in line with the services in Section 10. Hence, it is changed to Storage Commitment Service to be consistent with DIMSE.
3	**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? **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 *polling*[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. **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. **Decision**: As proposed.
3a	**Issue** [WG06]**:** do we only provide asynchronous or also synchronous response of the origin server? **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. **Decision:** As proposed.
3b	**Issue:** How to convey the (possible) failure codes as defined for storage commitment? **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). **Decision:** As proposed.
3c	**Issue:** What resources are used for requesting storage commitment? **Proposal:** Only one, the `commit` resource. See issue 4. **Decision:** As proposed.
3d	**Issue:** What different resource is to be used for polling for the result/checking the status of the request? **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). **Decision:** As proposed.
3e	**Issue:** What resource is being created as status resource? **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). **Decision:** As proposed.
3f	**Issue:** What resource is being created as result? **Proposal:** No resource is being created as result. The result will be given back as the payload of the response. Same advantage as before. **Decision:** As proposed.
3g	**Issue:** What kind of HTTP status codes are supported? **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. **Decision:** As proposed.

[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.

3h	**Issue:** Do we want to be able to cancel this long running request? **Proposal:** No as this is also not possible in DIMSE. **Decision:** As proposed.
4	**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. `/instance/{uid}/commit`). **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. 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. **Decision** (2022-09-19 WG06): Agree with proposal. If people want to commit per instance, they still can, using the chosen interface.
5	**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. **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. **Decision** (2022-09-19 WG06): Do not remove the Referenced SOP Class UID, as this would violate the proxy-ability.
6	**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)? **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). **Decision** (2022-09-16 WG06): As proposed, also for backwards compatibility.
7	**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. **Proposal**: No, this is no longer needed. In line with this, the optional Storage Media File-Set (U)ID are no longer possible. **Decision** (2022-09-19 WG06): keep in line with issue 5, so do return this too.
8	**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. **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). **Decision** (2022-09-16 WG06): as proposed; agreed with justification.
9	**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. **Proposal**: Yes, the user agent may add the Referenced Study and Series UIDs, and the Origin Server is required to accept them. **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.

	Decision (2022-09-16 WG06): as proposed; understood the justification. Applicable failure code will be (the already existing) 0112H – No such object instance.
10	**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. **Proposal**: The wrappers as described (from classic DICOM to DICOMweb) are relatively easy to create. 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 (un-wrappers?!) cannot be created. This is, however, not perceived as an issue. **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.
11	**Issue**: Should the Commit service also be in PS3.18, section 9. URI Service? **Proposal:** no, as store is not part of this service. **Decision** (2022-09-19 WG06): As proposed.
12	**Issue** [WG27]: Is it possible to make the Studies and Series references mandatory? **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. **Decision:** As proposed.
13	**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. **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. **Decision:** As proposed.
14	**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: • *Straightforward* – 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. • *Innovative* – 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 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). • *Hybrid* – 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). **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. **Decision:** As proposed.

15	**Issue**: How to support the SOP Class UID in the hierarchical data structures? How should it be part of the hierarchical payload. There are two options, namely a) to have it in as a separate layer of the hierarchy (this saves on data to be sent), or b) just have a Referenced SOP Sequence per Series (this saves on new attributes to be defined). **Proposal**: Have it in as a separate layer in the hierarchy. Data efficiency is a recurring gain, while definition is only a one-time cost. **Decision** (WG06, 2023-01-17): Do as proposed. Even though data efficiency is not the most important factor, the gain can be quite considerable, e.g. for a fMRI study containing 64K images per series. This motivates the decision.
16	**Issue:** Can storage commit in DICOMweb be solved in a (more) RESTful way, such that resources like series and instances can be set to be committed, and also be questioned about their commitment state? **Proposal:** While this is possible technically, there are several serious drawbacks. Such a pure approach generates more client-server traffic, is considerably less efficient in execution, and is harder to implement. Finally, it is also not possible to create bi-directional proxies with DIMSE services. **Decision** (WG06, 2023-01-17)**:** continue with currently chosen approach.
17	**Issue:** What Sequence and UID tags are to be used in the data structure? There are several tags referring to items that are needed: SOP Instance Sequences, SOP Class UIDs, Study UID sequences, etc. **Proposal:** let WG06 decide what are the most appropriate tags. **Decision** (2023-03-24 WG06)**:** use existing Referenced Study Sequence instead of proposed new Referenced Study Instance Sequence; likewise, use existing Referenced Series Sequence instead of new Referenced Series Instance Sequence; also, use existing Referenced Instance Sequence instead of new Referenced SOP Instance Sequence. Furthermore, use existing Study Instance UID instead of new Referenced Study Instance UID, and use Series Instance UID instead of new Referenced Series Instance UID.
18	**Issue**: Traditionally DICOMweb Service attribute optionality is defined in PS3.4 and referenced from PS3.18. Is it ok that we put it in Annex J in PS3.18 instead? The background of this question is that the available attributes have been extended intentionally in DICOMweb beyond what is in DIMSE. This makes it ‘awkward’ to put in PS3.4. Furthermore, we already have Annex I for the Store Instances Response Module, so we would follow an existing pattern. **Proposal**: Make it an Annex in PS3.18. **Decision** (WG06, 2023-08-28): Add a pointer to Annex J in PS3.4 (as there is a difference between DIMSE and DICOMweb); possibly a CP should be created for doing a similar reference to Annex I (when there is a difference).
19	**Issue**: Is the HTTP status code 404 Not Found the right number to convey that a Storage Commitment Request cannot be found at the origin server? **Proposal**: Yes. **Decision** (WG06, 2023-08-28): Add 410 Gone as another possible status code.

Scope and Field of Application

This supplement defines the means to perform storage commitment in DICOMweb. The Storage Commitment Service enables a user agent to arrange the safekeeping of Instances on an origin server.

The DICOMweb Storage Commitment Service is an extension to the existing DICOMweb services, mimicking the storage commitment service that is already available using DIMSE. Furthermore, it has been designed such that it is relatively easy to create proxies from/to DIMSE to/from DICOMweb Storage Commitment Service.

The DICOMweb variant of Storage Commit extends the DIMSE variant. In DICOMweb it is possible to provide the study and series context to the referenced instances; this provides more information for finding these instances at the server side.

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.

…

Add new section X Storage Commitment Service and Resources

X Storage Commitment Service and Resources

X.1 Overview

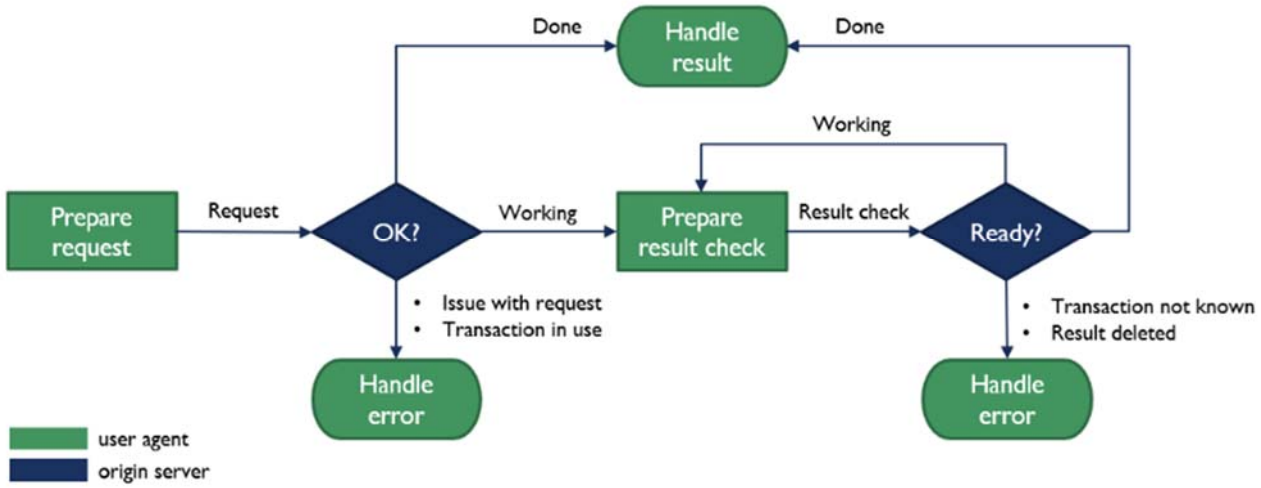
The Storage Commitment Service enables a user agent to request the safekeeping of Instances on an origin server. It corresponds to the DIMSE Storage Commitment Service Class as defined in Annex J of PS3.4 and has the same semantics.

As committing to storage of Instances is often a long-running operation on the origin server, the use of this service may be split into two transactions, at 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 storage commitment request it receives, for instance until the VNA that it uses for long term storage has given commitment for the referenced Instances.

Figure X.1-1 shows the possible scenarios of requesting storage commitment.

Figure X.1-1. Process of the Storage Commitment Service



This starts when the user agent sends a Request to the origin server. This requests the origin server's commitment to safekeep a set of SOP Instances, specified by their respective UIDs.

In case the origin server responds to the Request with Done, 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 Request with Working, it behaves asynchronously, and is working on the request. In this case, the user agent needs to perform a Result Check after some time. When this check is performed, the origin server may respond with Done, 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 respond to the Result Check with Working, which will trigger the user agent to perform a Result Check again. This process continues until the origin server responds with Done, finalizing the process.

For both the Request and the Result Check it is also possible that the origin server returns an error, and this also needs to be handled appropriately by the user agent; see Tables X.4.3-1 and X.5.3-1 for more details.

X.1.1 Resource Descriptions

There is one resource defined by this service:

/commitment-requests	Storage commitment requests managed by the origin server.

X.1.2 Common Query Parameters

The origin server shall support Query Parameters as required in Table X.1.2-1.

The user agent shall supply in the request Query Parameters as required in Table X.1.2-1.

Table X.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

See also Section 8.4.

X.1.3 Common Media Types

The origin server shall support the media types specified as Default or Required in Table X.1.3-1.

Table X.1.3-1. Default, Required, and Optional Media Types

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+json”	Required	Section 8.7.3.2
multipart/related; type=”application/dicom+xml”	Required	Section 8.7.3.2

X.2 Conformance

Implementations conforming to the Storage Commitment Service shall support the transactions listed as Required in Table X.2-1.

Table X.2-1. Required and Optional Transactions

Transaction	Support	Section
Request	Required	Section X.4
Result Check	Required	Section X.5

Implementations conforming to the Storage Commitment 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 Storage Commitment Service shall implement the Retrieve Capabilities Transaction, specifying its role (see Section 8.9 and Annex H).

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

An origin server implementation defines how it provides its commitment to storage. Certain origin servers may commit to permanently store the SOP Instances (e.g., an archive system) while other origin servers may commit to provide storage of the SOP Instances for a limited amount of time. The origin server shall document in its Conformance Statement the nature of its commitment to storage (e.g., duration of storage, retrieve capabilities and latency, capacity).

Once the origin server has commited to store the SOP Instances, the user agent may decide that it is appropriate to delete its copies of the SOP Instances. These types of behaviors are outside the scope of this Standard; however, the user agent shall document the types of behaviors it is able to provide in its Conformance Statement.

An origin server implementation shall specify in its Conformance Statement how long the result of a Request will be available for the user agent.

X.3 Transactions Overview

The Storage Commitment Service consists of the transactions listed in Table X.3-1.

Table X.3-1. Storage Commitment Service Transactions

Transaction Name	Method	Payload		Description
		Request	Success Response	
Request	POST	SOP Class UIDs and SOP Instance UIDs; optionally Study and Series UIDs	Storage Commitment Result	Requests to safekeep a referenced set of Instances.
Result Check	GET	N/A	Storage Commitment Result	Gets the result of a Request.

These transactions share the same resource (/commitment-requests/{transactionUID}) but are differentiated by their method.

X.4 Request Transaction

This transaction allows a user agent to request an origin server to commit to the safekeeping of a set of Instances.

X.4.1 Request

The request shall have the following syntax:

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

X.4.1.1 Target Resource

The Target Resource of this transaction is an individual commitment request identified by its Transaction UID.

X.4.1.2 Query Parameters

The request has no Query Parameters.

X.4.1.3 Request Header Fields

The origin server shall support Request Header Fields as required in Table X.4.1-2.

The user agent shall supply Request Header Fields as required in Table X.4.1-2.

Table X.4.1-2. 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.

X.4.1.4 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 for which the user agent requests the origin server to commit storage.

A request payload shall contain a Storage Commitment Request Module. See Annex J.1.

X.4.2 Behavior

The origin server shall process the storage commitment request. A success response either returns:

- a 200 (OK) status with a Storage Commitment Response payload that indicates the storage commitment status per referenced SOP Instance or
- a 202 (Accepted) status without payload indicating to the user agent that it should retrieve such a result later.

Note A 200 (OK) success status code should only be understood to mean that the request was successfully parsed and a Storage Commitment Response was returned by the origin server. The Storage Commitment Response may indicate that storage commitment failed for some or even all of the referenced SOP Instances.

X.4.3 Response

The response shall have the following syntax:

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

X.4.3.1 Status Codes

Table X.4.3-1 shows some common status codes corresponding to this transaction. See also Section 8.5 for additional status codes.

Table X.4.3-1. Status Code Meaning

Status	Code	Meaning
Success	200 (OK)	The origin server finished processing the storage commitment request; the payload describes in detail what referenced SOP Instances have been committed for safekeeping, and what instances have not.
	202 (Accepted)	The origin server has not finished processing the storage commitment request yet; there is no payload. The user agent is expected to follow-up with the Result Check transaction, as described in section X.5, to retrieve the result of the storage commitment request.
Failure	400 (Bad Request)	The origin server cannot handle the storage commitment request because of errors in the request headers or parameters.
	409 (Conflict)	The origin server cannot handle the storage commitment request because the provided transaction UID is already in use.
	503 (Service Unavailable)	The origin server cannot handle the storage commitment request; this may be a temporal or permanent state.

X.4.3.2 Response Header Fields

The origin server shall support header fields as required in Table X.4.3-2.

Table X.4.3-2. Response Header Fields

Name	Value	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.
Retry-After	uint	O	The number of seconds the user agent is requested to wait until a (next) result check or retrying the request.

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
```

X.4.3.3 Response Payload

A 200 (OK) success response payload shall contain a Storage Commitment Response Module. See Annex J.2.

A 202 (Accepted) success response will not contain a payload.

Any failure response payload may contain a Status Report describing failures, warnings, or other useful information.

X.5 Result Check Transaction

This transaction allows a user agent to request an origin server to provide the result of an earlier Request.

Note The user agent uses this transaction when the origin server has responded with status code 202 (Accepted) to either a Request or a Result Check transaction.

X.5.1 Request

The request shall have the following syntax:

```
GET SP /commitment-requests/{transactionUID} SP version CRLF
Accept: 1#media-type CRLF
*(header-field CRLF)
CRLF
```

X.5.1.1 Target Resource

The Target Resource of this transaction is an individual commitment request identified by its Transaction UID.

X.5.2.2 Query Parameters

The request has no Query Parameters.

X.5.2.3 Request Header Fields

The origin server shall support Result Check Header Fields as required in Table X.5.1-2.

The user agent shall supply Result Check Header Fields as required in Table X.5.1-2.

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

Table X.5.1-2. 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.

X.5.1.4 Request Payload

The request has no payload.

X.5.2 Behavior

If the result identified by the Transaction UID is available on the origin server, this result is returned in an Acceptable Media Type (see section 8.7.4); the result contains in detail what referenced SOP Instances have been committed for safekeeping, and what instances have not.

If this result is not yet available, the server will return that it is still working on the storage commitment request.

X.5.3 Response

The response shall have the following syntax:

```
version SP status-code SP reason-phrase CRLF
[retry after 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 finished processing the Request transaction identified by the supplied Transaction UID (see X.4); the payload contains the result.
	202 (Accepted)	The origin server has not yet finished processing the Request transaction identified by the supplied Transaction UID; there is no payload. The user agent is expected to follow-up again with the Result Check transaction, to retrieve the result of the storage commitment request.
Failure	404 (Not Found)	The origin server cannot find the storage commitment request result identified by the supplied Transaction UID.

	410 (Gone)	The origin server can no longer provide the storage commitment request result identified by the supplied Transaction UID.
	503 (Service Unavailable)	The origin server cannot handle the Result Check request; this may be a temporary or permanent state.

Note The 'Not Found' status code may be caused by a wrong Transaction UID has been supplied by the user agent, or that the origin server has deleted the applicable result.

Note The "Gone' status code may be caused by the origin server deleting the applicable result, but still having a record of the Transaction UID.

Note When the 'Not Found' or the 'Gone' status code is returned, the user agent might initiate a new storage commitment request. When the 'Service Unavailable' status code is returned, the user agent might retry later with another Result Check transaction.

X.5.3.2 Response Header Fields

See Section X.4.3.2.

10.x.5.3 Response Payload

See Section X.4.3.3.

Add new examples for the Storage Commitment Service

B Examples (Informative)

…

B.x1 Request Storage Commitment for Multiple Instances with JSON

This example shows the flow of messages between the user agent and the origin server for the scenario in which 1) the user agent requests storage commitment for two SOP Instances in JSON, 2) the origin server tells the user agent to check for the result of this request later, 3) the user agent checks for the result, and 4) the result provided by the origin server shows that it commits to safely store one of the instances, while it does not commit to safely store the other instance.

Step 1 of this scenario involves the user agent sending a POST request for the two instances with transaction UID 1.1.99999.20220901 in the HTTP header:

```
POST /radiology/commitment-requests/1.1.99999.20220901 HTTP/1.1
Host: www.hospital-stmarco
Content-Type: application/dicom+json
…
{
  "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"
          ]
        }
      }]
  }
}
```

Here, the references to the applicable SOP instances are in the Referenced SOP Sequence (0008,1199); see Table J.1-1 for the possible structures of the storage commitment request. The SOP Class UID of both the instances is CT Image (for both instances attribute (0008,1150) has value 1.2.840.10008.5.1.4.1.1.2), and the applicable instances are identified by their respective SOP Instance UIDs (the values of (0008,1155) are 1.3.12.2.1107.5.99.3.30000012031310075961300000059 and ….00060) respectively).

In step 2 the origin server returns its response to the request. In this scenario this is the asynchronous case where there is no immediate result (return code 202 Accepted), and where the server also notifies the user agent that it ought to wait at least 300 seconds before making a follow-up request for the result; the synchronous response case would skip steps 2 and 3, and would continue at step 4.

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

In step 3, after waiting the suggested period of time, the user agent GETs the status of the request using the same transaction UID as given in the original request:

```
GET /radiology/commitment-requests/1.1.99999.20220901 HTTP/1.1
Host: www.hospital-stmarco
Content-Type: application/dicom+json
…
```

Step 4 of this scenario involves the origin server returning the result of the storage commitment request. In this case it is the response to the result check as shown in step 3. Note that in case the server initially responds to the POST request of step 1 with the HTTP response status code 200 (the synchronous case) the same result would be returned:

```
HTTP/1.1 200 OK
Content-Length: 842
Content-Type: application/dicom+json; charset=utf-8

{
  "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 ]
        }
      }]
    }
}
```

The origin server provided 274 as value of the failure reason (0008,1197). This is 0112H and means “No such object instance” (see section C.14.1.1 of Part 3). Apparently, the SOP Instance identified by SOP Instance UID 1.3.12.2.1107.5.99.3.30000012031310075961300000060 is not on the origin server.

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

The intent of this example is the same as presented in B.x1, namely the scenario to request storage commitment for two SOP Instances, where for one it succeeds, and for one it fails. The differences are in the synchronicity (in this case it is synchronous), the syntax (in this case using XML), and the structure (in this case using the hierarchical study-series-SOP Class-instance structure, starting with a Referenced Study Sequence (0008,1110); see Table J.1-1 for more details on this structure).

Step 1:

```
POST /radiology/commitment-requests/1.1.99999.20220901 HTTP/1.1
Host: www.hospital-stmarco
Content-Type: application/dicom+xml

<?xml version="1.0" encoding="UTF-8"?>
<NativeDicomModel>
  <DicomAttribute Tag="00081110" VR="SQ" Keyword="ReferencedStudySequence">
  <Item number="1">
    <DicomAttribute Tag="0020000D" VR="UI" Keyword="StudyInstanceUID">
      <Value number="1">1.2.250.1.59.40211.12345678.678910</Value>
    </DicomAttribute>
    <DicomAttribute Tag="00081115" VR="SQ" Keyword="ReferencedSeriesSequence">
    <Item number="1">
      <DicomAttribute Tag="0020000E" VR="UI" Keyword="SeriesInstanceUID">
        <Value number="1">1.2.250.1.59.40211.789001276.14556172.67789</Value>
      </DicomAttribute>
      <DicomAttribute Tag="0008XXX1" VR="SQ" Keyword="ReferencedSOPClassSequence">
      <Item number="1">
        <DicomAttribute Tag="00081150" VR="UI" Keyword="ReferencedSOPClassUID">
          <Value number="1">1.2.840.10008.5.1.4.1.1.2</Value>
            </DicomAttribute>
        <DicomAttribute Tag="0008114A" VR="SQ" Keyword="ReferencedInstanceSequence">
        <Item number="1">
          <DicomAttribute Tag="00081155" VR="UI" Keyword="ReferencedSOPInstanceUID">
```

```
              <Value number="1">1.3.12.2.1107.5.99.3.30000012031310075961300000059</Value>
            </DicomAttribute>
          </Item>
          <Item number="2">
            <DicomAttribute Tag="00081155" VR="UI" Keyword="ReferencedSOPInstanceUID">
              <Value number="1">1.3.12.2.1107.5.99.3.30000012031310075961300000060</Value>
            </DicomAttribute>
          </Item>
          </DicomAttribute>
        </Item>
        </DicomAttribute>
      </Item>
      </DicomAttribute>
    </Item>
    </DicomAttribute>
</NativeDicomModel>
```

In the above, the applicable study is identified by its Study Instance UID (0020,000D) with value 1.2.250.1.59.40211.12345678.678910. The applicable series is identified by its Series Instance UID (0020,000E) with value 1.2.250.1.59.40211.789001276.14556172.67789. SOP Class UIDs and SOP Instance UIDs are the same as the example given in B.x1.

Step 2:

```
HTTP/1.1 200 OK
Content-Length: 2879
Content-Type: application/dicom+xml

<?xml version="1.0" encoding="UTF-8"?>
<NativeDicomModel>
  <DicomAttribute Tag="00081110" VR="SQ" Keyword="ReferencedStudySequence">
  <Item number="1">
    <DicomAttribute Tag="0020000D" VR="UI" Keyword="StudyInstanceUID">
      <Value number="1">1.2.250.1.59.40211.12345678.678910</Value>
    </DicomAttribute>
    <DicomAttribute Tag="00081115" VR="SQ" Keyword="ReferencedSeriesSequence">
    <Item number="1">
      <DicomAttribute Tag="0020000E" VR="UI" Keyword="SeriesInstanceUID">
        <Value number="1">1.2.250.1.59.40211.789001276.14556172.67789</Value>
      </DicomAttribute>
      <DicomAttribute Tag="0008XXX1" VR="SQ" Keyword="ReferencedSOPClassSequence">
      <Item number="1">
        <DicomAttribute Tag="00081150" VR="UI" Keyword="ReferencedSOPClassUID">
             <Value number="1">1.2.840.10008.5.1.4.1.1.2</Value>
           </DicomAttribute>
        <DicomAttribute Tag="0008114A" VR="SQ" Keyword="ReferencedInstanceSequence">
        <Item number="1">
          <DicomAttribute Tag="00081155" VR="UI" Keyword="ReferencedSOPInstanceUID">
            <Value number="1">1.3.12.2.1107.5.99.3.30000012031310075961300000059</Value>
          </DicomAttribute>
        </Item>
        </DicomAttribute>
      </Item>
      </DicomAttribute>
    </Item>
    </DicomAttribute>
  </Item>
  </DicomAttribute>
  <DicomAttribute Tag="0008XXX2" VR="SQ" Keyword="FailedStudySequence">
  <Item number="1">
    <DicomAttribute Tag="0020000D" VR="UI" Keyword="StudyInstanceUID">
      <Value number="1">1.2.250.1.59.40211.12345678.678910</Value>
    </DicomAttribute>
    <DicomAttribute Tag="00081115" VR="SQ" Keyword="ReferencedSeriesSequence">
    <Item number="1">
      <DicomAttribute Tag="0020000E" VR="UI" Keyword="SeriesInstanceUID">
        <Value number="1">1.2.250.1.59.40211.789001276.14556172.67789</Value>
      </DicomAttribute>
      <DicomAttribute Tag="0008XXX1" VR="SQ" Keyword="ReferencedSOPClassSequence">
```

```
        <Item number="1">
          <DicomAttribute Tag="00081150" VR="UI" Keyword="ReferencedSOPClassUID">
            <Value number="1">1.2.840.10008.5.1.4.1.1.2</Value>
             </DicomAttribute>
          <DicomAttribute Tag="0008114A" VR="SQ" Keyword="ReferencedInstanceSequence">
          <Item number="1">
            <DicomAttribute Tag="00081155" VR="UI" Keyword="ReferencedSOPInstanceUID">
              <Value number="1">1.3.12.2.1107.5.99.3.30000012031310075961300000060</Value>
            </DicomAttribute>
            <DicomAttribute Tag="00081197" VR="UI" Keyword="FailureReason">
              <Value number="1">274</Value>
            </DicomAttribute>
          </Item>
          </DicomAttribute>
        </Item>
        </DicomAttribute>
      </Item>
      </DicomAttribute>
    </Item>
    </DicomAttribute>
  </NativeDicomModel>
```

The Failed Study Sequence (0008,xxx2) has the same structure as the Referenced Study Sequence, except that it adds a Failure Reason (0008,1197) to each Referenced SOP Instance UID, indicating why the origin server could not commit to safely store that referenced SOP Instance.

B.x3 Request Storage Commitment with HTTP Multipart Request for Instances from Multiple Studies

This example has the same intent as B.x2, but differs in having a multipart request, and the SOP Instances belonging to different studies.

Step 1:

```
POST /radiology/commitment-requests/1.1.99999.20220901 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="00081110" VR="SQ" Keyword="ReferencedStudySequence">
  <Item number="1">
    <DicomAttribute Tag="0020000D" VR="UI" Keyword="StudyInstanceUID">
      <Value number="1">1.2.250.1.59.40211.12345678.678910</Value>
    </DicomAttribute>
    <DicomAttribute Tag="00081115" VR="SQ" Keyword="ReferencedSeriesSequence">
    <Item number="1">
      <DicomAttribute Tag="0020000E" VR="UI" Keyword="SeriesInstanceUID">
        <Value number="1">1.2.250.1.59.40211.789001276.14556172.67789</Value>
      </DicomAttribute>
      <DicomAttribute Tag="0008XXX1" VR="SQ" Keyword="ReferencedSOPClassSequence">
      <Item number="1">
        <DicomAttribute Tag="00081150" VR="UI" Keyword="ReferencedSOPClassUID">
          <Value number="1">1.2.840.10008.5.1.4.1.1.2</Value>
        </DicomAttribute>
        <DicomAttribute Tag="0008114A" VR="SQ" Keyword="ReferencedInstanceSequence">
        <Item number="1">
          <DicomAttribute Tag="00081155" VR="UI" Keyword="ReferencedSOPInstanceUID">
            <Value number="1">1.3.12.2.1107.5.99.3.30000012031310075961300000059</Value>
          </DicomAttribute>
        </Item>
        </DicomAttribute>
      </Item>
      </DicomAttribute>
    </Item>
    </DicomAttribute>
  </Item>
```

```
  </DicomAttribute>
</NativeDicomModel>
--MESSAGEBOUNDARY
<?xml version="1.0" encoding="UTF-8"?>
<NativeDicomModel>
  <DicomAttribute Tag="00081110" VR="SQ" Keyword="ReferencedStudySequence">
  <Item number="1">
    <DicomAttribute Tag="0020000D" VR="UI" Keyword="StudyInstanceUID">
      <Value number="1">1.2.250.1.59.40211.12345678.678911</Value>
    </DicomAttribute>
    <DicomAttribute Tag="00081115" VR="SQ" Keyword="ReferencedSeriesSequence">
    <Item number="1">
      <DicomAttribute Tag="0020000E" VR="UI" Keyword="SeriesInstanceUID">
        <Value number="1">1.2.250.1.59.40211.789001276.14556172.68856</Value>
      </DicomAttribute>
      <DicomAttribute Tag="0008XXX1" VR="SQ" Keyword="ReferencedSOPClassSequence">
      <Item number="1">
        <DicomAttribute Tag="00081150" VR="UI" Keyword="ReferencedSOPClassUID">
          <Value number="1">1.2.840.10008.5.1.4.1.1.2</Value>
        </DicomAttribute>
        <DicomAttribute Tag="0008114A" VR="SQ" Keyword="ReferencedInstanceSequence">
        <Item number="1">
          <DicomAttribute Tag="00081155" VR="UI" Keyword="ReferencedSOPInstanceUID">
            <Value number="1">1.3.12.2.1107.5.99.3.30000012031310075961300000060</Value>
          </DicomAttribute>
        </Item>
        </DicomAttribute>
      </Item>
      </DicomAttribute>
    </Item>
    </DicomAttribute>
  </Item>
  </DicomAttribute>
</NativeDicomModel>
--MESSAGEBOUNDARY
```

Step 2:

```
HTTP/1.1 200 OK
Content-Length: 2801
Content-Type: application/dicom+xml

<?xml version="1.0" encoding="UTF-8"?>
<NativeDicomModel>
  <DicomAttribute Tag="00081110" VR="SQ" Keyword="ReferencedStudySequence">
  <Item number="1">
    <DicomAttribute Tag="0020000D" VR="UI" Keyword="StudyInstanceUID">
      <Value number="1">1.2.250.1.59.40211.12345678.678910</Value>
    </DicomAttribute>
    <DicomAttribute Tag="00081115" VR="SQ" Keyword="ReferencedSeriesSequence">
    <Item number="1">
      <DicomAttribute Tag="0020000E" VR="UI" Keyword="SeriesInstanceUID">
        <Value number="1">1.2.250.1.59.40211.789001276.14556172.67789</Value>
      </DicomAttribute>
      <DicomAttribute Tag="0008XXX1" VR="SQ" Keyword="ReferencedSOPClassSequence">
      <Item number="1">
        <DicomAttribute Tag="00081150" VR="UI" Keyword="ReferencedSOPClassUID">
              <Value number="1">1.2.840.10008.5.1.4.1.1.2</Value>
        </DicomAttribute>
        <DicomAttribute Tag="0008114A" VR="SQ" Keyword="ReferencedInstanceSequence">
        <Item number="1">
          <DicomAttribute Tag="00081155" VR="UI" Keyword="ReferencedSOPInstanceUID">
            <Value number="1">1.3.12.2.1107.5.99.3.30000012031310075961300000059</Value>
          </DicomAttribute>
        </Item>
        </DicomAttribute>
      </Item>
      </DicomAttribute>
    </Item>
    </DicomAttribute>
  </Item>
```

```
    <Item number="2">
      <DicomAttribute Tag="0020000D" VR="UI" Keyword="StudyInstanceUID">
        <Value number="1">1.2.250.1.59.40211.12345678.678911</Value>
      </DicomAttribute>
      <DicomAttribute Tag="00081115" VR="SQ" Keyword="ReferencedSeriesSequence">
      <Item number="1">
        <DicomAttribute Tag="0020000E" VR="UI" Keyword="SeriesInstanceUID">
          <Value number="1">1.2.250.1.59.40211.789001276.14556172.68856</Value>
        </DicomAttribute>
        <DicomAttribute Tag="0008XXX1" VR="SQ" Keyword="ReferencedSOPClassSequence">
        <Item number="1">
          <DicomAttribute Tag="00081150" VR="UI" Keyword="Referenced SOP Class UID">
            <Value number="1">1.2.840.10008.5.1.4.1.1.2</Value>
          </DicomAttribute>
          <DicomAttribute Tag="0008114A" VR="SQ" Keyword="ReferencedInstanceSequence">
          <Item number="1">
            <DicomAttribute Tag="00081155" VR="UI" Keyword="ReferencedSOPInstanceUID">
              <Value number="1">1.3.12.2.1107.5.99.3.30000012031310075961300000060</Value>
            </DicomAttribute>
            <DicomAttribute Tag="00081197" VR="UI" Keyword="FailureReason">
              <Value number="2">274</Value>
            </DicomAttribute>
          </Item>
          </DicomAttribute>
        </Item>
        </DicomAttribute>
      </Item>
      </DicomAttribute>
    </Item>
    </DicomAttribute>
</NativeDicomModel>
```

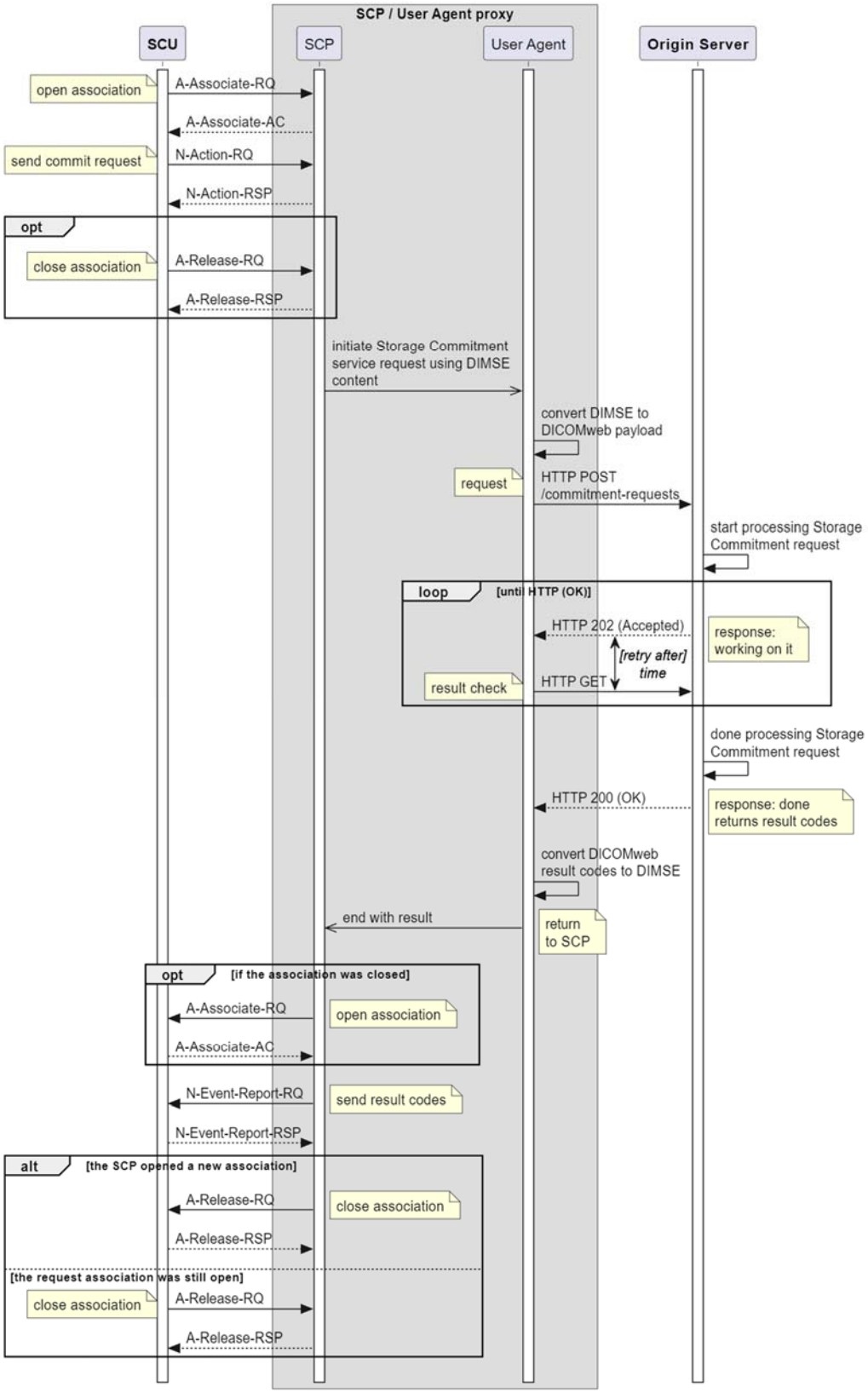
B.x4 Bi-directional Proxy for Storage Commitment

The DICOMweb Storage Commitment Service may be deployed in a hybrid environment, i.e. an environment in which both DICOMweb and DIMSE are used. In such a hybrid environment, a proxy can broker transactions from one service to the other, allowing a DICOMweb origin server or a DIMSE SCP to support storage commitment for a mixed set of DICOMweb user agents and DIMSE SCUs.

DICOM does not require an implementation of proxies; however, since they would be very useful in a hybrid environment, the examples in this section show how this could be done.

Figure B.x4-1 shows how a proxy could facilitate a request for Storage Commitment from a DIMSE SCU to a DICOMweb origin server.

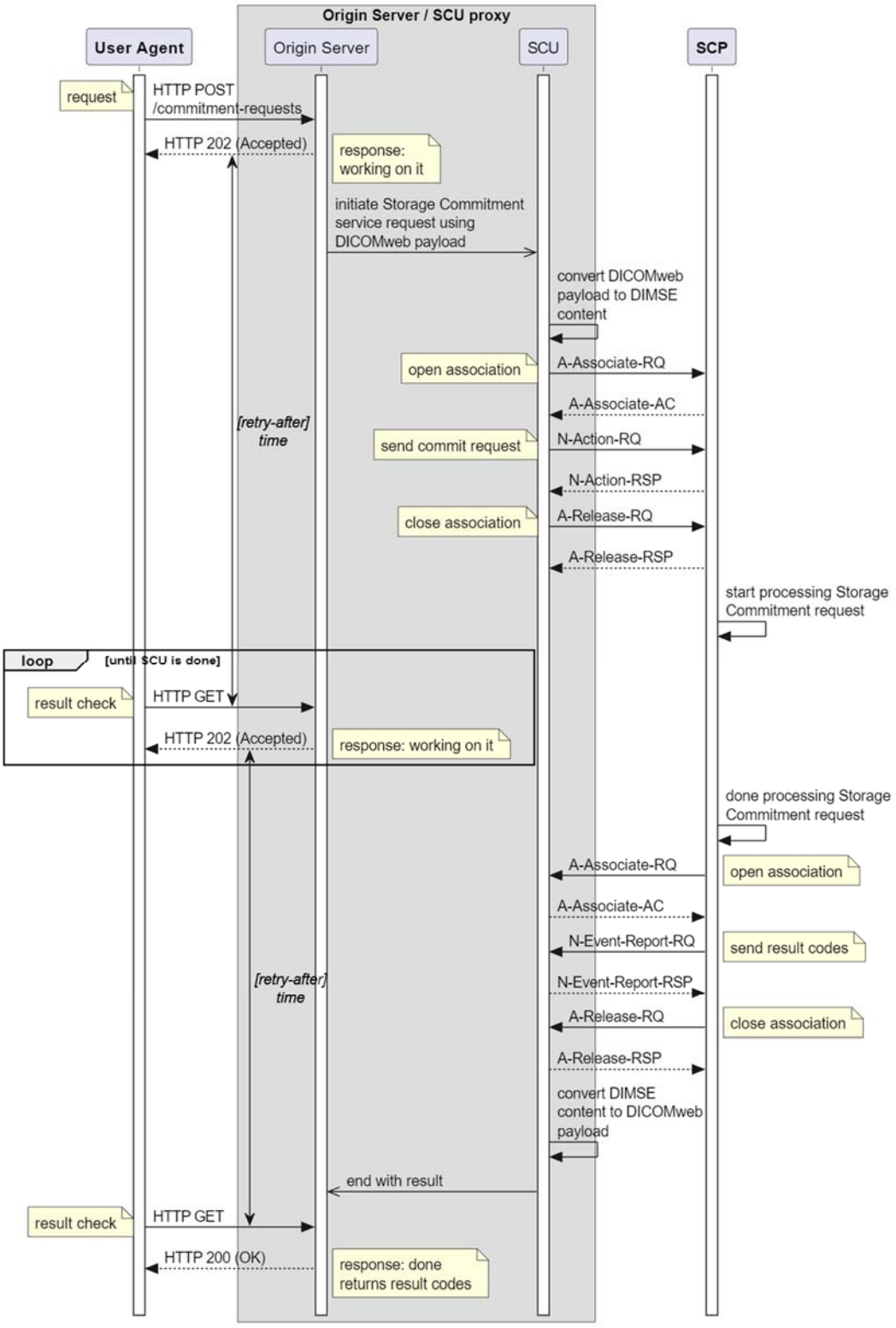
620 **Figure B.x4-1. Storage Commitment DIMSE Proxy for a DICOMweb Origin Server**



621

Figure B.x4-2 shows how a proxy could facilitate a request for Storage Commitment from a DICOMweb user agent to a DIMSE SCP.

Figure B.x4-2. Storage Commitment DICOMweb Proxy for a DIMSE SCP



626 ***Update Table H-1 Resources and Methods***

627 # H Capabilities Description

628 …

629 **Table H-1. Resources and Methods**

Service	Resource	Transactions	Reference
Studies (see Section 10.1.1)			
	studies	Search for Studies Store Instances	Section 10.6 Section 10.5
	{StudyInstance}	Retrieve Study Store Study Instances	Section 10.4 Section 10.5
	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
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
Storage Commitment Requests (see Section X.1.1)			
	commitment-requests	**Request**	**Section X.4**
		Result Check	**Section X.5**

630

631 ***Add new Annex J Storage Commitment Modules***

632 # J Storage Commitment Modules

633 ## J.1 Storage Commitment Request Module

634 Table J.1-1 specifies the Attributes of the Storage Commitment Request Module.

635 **Table J.1-1. Storage Commitment Request Module**

Attribute Name	Tag	Type	Attribute Description
Referenced SOP Sequence	(0008,1199)	1C	The SOP Instances for which storage commitment is requested. One or more Items shall be included in this Sequence. Required if the Referenced Study Sequence (0008,1110) is absent.
>Table 10-11 "SOP Instance Reference Macro Attributes" in PS3.3			
Referenced Study Sequence	(0008,1110)	1C	The Studies containing Instances for which storage commitment is requested. One or more Items shall be included in this Sequence. Required if the Referenced SOP Sequence (0008,1199) is absent.
>Study Instance UID	(0020,000D)	1	Uniquely identifies the referenced Study.
>Referenced Series Sequence	(0008,1115)	1	The Series containing Instances for which storage commitment is requested. One or more Items shall be included in this Sequence.
>>Series Instance UID	(0020,000E)	1	Uniquely identifies the referenced Series.
>>Referenced Instances by SOP Class Sequence	(0008,xxx1)	1	The SOP Instances for which storage commitment is requested grouped by SOP Class. One or more Items shall be included in this Sequence.
>>>Referenced SOP Class UID	(0008,1150)	1	Uniquely identifies the referenced SOP Class.
>>>Referenced Instance Sequence	(0008,114A)	1	The SOP Instances for which storage commitment is requested. One or more Items shall be included in this Sequence.
>>>>Referenced SOP Instance UID	(0008,1155)	1	Uniquely identifies the referenced SOP Instance.

636

637 ## J.2 Storage Commitment Reply Module

638 Table J.2-1 specifies the Attributes of the Storage Commitment Response Module.

639 **Table J.2-1. Storage Commitment Response Module**

Attribute Name	Tag	Type	Attribute Description
Referenced SOP Sequence	(0008,1199)	1C	The SOP Instances for which storage has been committed. Required if the request payload contained the Referenced SOP Sequence (0008,1199), and there is at least one SOP Instance in that supplied sequence for which storage has been committed.
>Table 10-11 "SOP Instance Reference Macro Attributes" in PS3.3			
Failed SOP Sequence	(0008,1198)	1C	The SOP Instances for which storage has not been committed. Required if the request payload contained the Referenced SOP Sequence (0008,1199), and there is at least one SOP Instance in that supplied sequence for which storage has not been committed.
>Table 10-11 "SOP Instance Reference Macro Attributes" in PS3.3			
>Failure Reason	(0008,1197)	1	The reason that storage has not been committed for this SOP Instance. See PS3.3, section C.14.1.1 for possible values.
Referenced Study Sequence	(0008,1110)	1C	The Studies containing Instances for which storage has been committed. Required if the request payload contained the Referenced Study Sequence (0008,1110), and there is at least one SOP Instance for which storage has been committed
>Study Instance UID	(0020,000D)	1	Uniquely identifies the referenced Study.
>Referenced Series Sequence	(0008,1115)	1	The Series containing Instances for which storage has been committed.
>>Series Instance UID	(0020,000E)	1	Uniquely identifies the referenced Series.
>>Referenced Instances by SOP Class Sequence	(0008,xxx1)	1	The SOP Instances for which storage has been committed grouped by SOP Class.
>>>Referenced SOP Class UID	(0008,1150)	1	Uniquely identifies the referenced SOP Class.
>>>Referenced Instance Sequence	(0008,114A)	1	The SOP Instances for which storage has been committed.
>>>>Referenced SOP Instance UID	(0008,1155)	1	Uniquely identifies the referenced SOP Instance.

Failed Study Sequence	(0008,xxx2)	1C	The Studies containing Instances for which storage has not been committed. Required if the request payload contained the Referenced Study Sequence (0008,1110), and there is at least one SOP Instance in that supplied sequence for which storage has not been committed.
>Study Instance UID	(0020,000D)	1	Uniquely identifies the referenced Study.
>Referenced Series Sequence	(0008,1115)	1	The Series containing Instances for which storage has not been committed.
>>Referenced Series Instance UID	(0020,000E)	1	Uniquely identifies the referenced Series.
>>Referenced Instances by SOP Class Sequence	(0008,xxx1)	1	The SOP Instances for which storage has not been provided grouped by SOP Class.
>>>Referenced SOP Class UID	(0008,1150)	1	Uniquely identifies the referenced SOP Class.
>>>Referenced Instance Sequence	(0008,114A)	1	The SOP Instances for which storage has not been committed.
>>>>Referenced SOP Instance UID	(0008,1155)	1	Uniquely identifies the referenced SOP Instance.
>>>>Failure Reason	(0008,1197)	1	The reason that storage has not been committed for this SOP Instance.

640

641

Changes to NEMA Standards Publications PS 3.2

Add new subsection to N.1.3.

N DICOM Conformance Statement Template (Normative)

…

N.1 Overview

…

N.1.3 DICOM Web Services

…

N.1.3.x Storage Commitment Service

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

[Complete Table 0- to indicate support for the Storage Commitment Web Service]

Table 0-x1 Storage Commitment Service

Service	Transaction	Resource	User Agent	Origin Server
Storage Commitment Service	*Request*	commitment-requests		
	Result Check	commitment-requests		

Add a new subsection on the Storage Commitment Service to section N.5.3 Supported DICOM Web Services

N.5 Service and Interoperability Description

…

N.5.3 Supported DICOM Web Services

…

N.5.3.x Storage Commitment Web Service

This section provides details regarding the Storage Commitment Web Service. For an overview of supported Transactions and resources see Table 0-xx1 Storage Commitment Service.

N.5.3.x.1 Request Transaction – Storage Commitment Service

N.5.3.x.1.1 User Agent

The Request Transaction user agent can request resources listed in Table N.5-xx1.

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

Table N.5-xx1: Resources for Request Transaction – User Agent

Resource	Comments
	See resource path in section X.1.1 in PS3.18
Commitment-requests	

The Request Transaction user agent supports Header Fields listed in Table N.5-xx2.

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

Table N.5-xx2: Header Fields for Request 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		*[If Content-Encoding is not present]*
Content-Encoding		*[If Content-Length is not present]*

N.5.3.x.1.2 Origin Server

The Request Transaction origin server receives POST requests for storage commitment of the referenced 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 an appropriate Content-Type (i.e. XML or JSON).

The URI is composed by a Base URI: See Base URI for the origin server in Section N.6.3.x.

The Request Transaction origin server supports resources listed in Table N.5-xx3.

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

Table N.5-xx3: Resources for Request Transaction – Origin Server

Resource	Comments
	See resource path in section X.1.1 in PS3.18
commitment-requests	

The Request Transaction origin server supports Header Fields listed in Table N.5-xx4.

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

Table N.5-xx4: Header Fields for Request 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		*[If Content-Encoding is not present]*
Content-Encoding		*[If Content-Length is not present]*

691

692 **N.5.3.x.2 Result Check Transaction – Storage Commitment Service**

693 **N.5.3.x.2.1 User Agent**

694 The resources and header fields supported by the user agent for the Result Check Transaction are the
695 same as for the Request Transaction; see section N.5.3.x.1.1.

696 **N.5.3.x.2.2 Origin Server**

697 The Result Check Transaction origin server receives GET requests to check whether there is a result for
698 a storage commitment request.

699 The Base URI, resources, and header fields supported by the origin server for the Result Check Transac-
700 tion are the same as for the Request Transaction; see section N.5.3.x.1.2.

701 ***Add a new subsection on the Storage Commitment Service to section N.6.3 Configuration of DI-***
702 ***COM Web Services.***

703 …

704 **N.6 Configuration**

705 …

706 **N.6.3 Configuration of DICOM Web Services**

707 …

708 **N.6.3.x Storage Commitment Service Configuration**

709 **N.6.3.x.1 Request Transaction Configuration**

710 Table N.6-xx5 lists configuration parameters for the Request Transaction of the Storage Commitment
711 Service:

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

713 **Table N.6-xx5: Request and Result Check Transaction Parameters**

Local Configuration Parameters – Request and Result Check Transaction			
Parameter	**Configurable**	**Default Value**	**Comments**
	<<USER SERVICE FIXED>>	*[If there is no default, leave blank]*	*[Provide comments or Values/ranges if applicable]*
Commit local Origin Server URL (Base URI)	*FIXED*	*http://<host-name>:<port>/commitment-requests*	
Port	*SERVICE*	8081	
Secured Commit local Origin Server URL (Base URI)	*SERVICE*		

Secured Port	*SERVICE*		
Result Availability Duration	*FIXED*	*24*	*The number of hours that the storage commitment request result is guaranteed to be retrievable from the origin server.*
<Specific Storage Commitment Service parameter>			
Remote Configuration Parameters – Request and Result Check Transaction			
[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.]			
Parameter	**Configurable**	**Default Value**	**Comments**
	<<USER SERVICE FIXED>>	*[If there is no default, leave blank]*	*[Provide comments or Values/ranges if applicable]*
Commit remote Origin Server URL	*USER*		
Port	*USER*		
Secured Commit Remote Origin Server URL	*SERVICE*		
Secured Port	*SERVICE*		
Result Availability Duration	*FIXED*	*24*	*The number of hours that the storage commitment request result is guaranteed to be retrievable from the origin server.*
<Specific Storage Commitment Service parameter>			

N.6.3.x.2 Result Check Transaction Configuration

Table N.6-xx5 lists configuration parameters for the Result Check Transaction of the Storage Commitment Service.

Add a new subsection on the Storage Commitment Service to section N.7.3.3 DICOM Web Services.

N.7 Network and Media Communication Details

…

N.7.3 Status Codes

…

724 **N.7.3.3 DICOM Web Services**

725 …

726 **N.7.3.3.x Storage Commitment Service**

727 **N.7.3.3.x.1 Request Transaction as Origin Server**

728 Table N.7-xx6 lists the Status Codes that an origin server supports for the Request Transaction of the
729 Storage Commitment Service and the condition in which any of the listed Status Codes is sent.

730 *[Describe below the condition in which the application sends the specific Status Codes in the Request Transaction*
731 *response as origin server.]*

732 **Table N.7-xx6: Status Codes of Origin Server for Request Transaction**

Status	Code	Condition
Success	*200 (OK)*	*The origin server finished processing the storage commitment request*
	202 (Accepted)	*The origin server has not finished processing the storage commitment request yet*
Failure	*400 (Bad Request)*	*The origin server cannot handle the storage commitment request because of errors in the request headers or parameters*
	409 (Conflict)	*The origin server cannot handle the storage commitment request because the provided transaction UID is already in use*
	503 (Service Unavailable)	*The origin server cannot handle the storage commitment request; this may be a temporal or permanent state*

733

734 **N.7.3.3.x.2 Request Transaction as User Agent**

735 Table N.7-xx7 lists the Status Codes that a user agent supports for the Request Transaction of the Stor-
736 age Commitment Service and defines the application behavior, when encountering any of the listed Sta-
737 tus Codes.

738 *[Describe below the behavior of the application when it receives various Status Codes in the Request Transaction*
739 *response]*

740 **Table N.7-xx7: Status Codes of User Agent for Request Transaction**

Status	Code	Behavior
Success	*200 (OK)*	*Mark all SOP Instances for which the origin server committed safe storage as ready for deletion*
	202 (Accepted)	*Retry later to get the result of the request*
Failure	*400 (Bad Request)*	*Reformat the request to proper HTTP*
	409 (Conflict)	*Retry with another transaction UID*
*	*Any other code*	*Do further analysis*

741

742 **N.7.3.3.x.3 Result Check Transaction as Origin Server**

743 Table N.7-xx8 lists the Status Codes that an origin server supports for the Result Check Transaction of
744 the Storage Commitment 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 Result Check Transaction response as origin server.]

Table A.7-xx8: Status Codes of Origin Server for Result Check Transaction

Status	Code	Condition
Success	*200 (OK)*	*The origin server finished processing the storage commitment request*
	202 (Accepted)	*The origin server has not finished processing the storage commitment request yet*
Failure	*404 (Not Found)*	*The origin server cannot find the storage commitment request result*
	410 (Gone)	*The origin server can no longer provide the storage commitment request result*
	503 (Service Unavailable)	*The origin server cannot handle the result check request; this may be a temporary or permanent state*

N.7.3.3.x.4 Result Check Transaction as User Agent

Table N.7-xx9 lists the Status Codes that a user agent supports for the Result Check Transaction of the Storage Commitment Service and 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 Result Check Transaction response]

Table N.7-xx9: Status Codes of User Agent for Result Check Transaction

Status	Code	Behavior
Success	*200 (OK)*	*Mark all SOP Instances for which the origin server committed safe storage as ready for deletion*
	202 (Accepted)	*Retry later to get the result of the request*
Failure	*404 (Not Found)*	*Start all over with a storage commitment request*
	410 (Gone)	*Start all over with a storage commitment request*
*	*Any other code*	*Do further analysis*

Changes to NEMA Standards Publications PS 3.4

Adapt the text on DIMSE Storage Commitment in section J.1.1 to be consistent with this supplement.

J.1.1 Scope

…

Once the SCP has ~~**accepted the**~~ commit~~**men**~~t**ed** to store the SOP Instances, the SCU may decide that it is appropriate to delete its copies of the SOP Instances. These types of ~~**policies**~~**behaviors** are outside the scope of this Standard, however, the SCU is required to document these ~~**policies**~~**behaviors** in its Conformance Statement.

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)	**Referenced Instances by SOP Class Sequence**	**ReferencedInstanceBy-SOPClassSequence**	**SQ**	**1**	
(0008,xxx2)	**Failed Study Sequence**	**FailedStudySequence**	**SQ**	**1**	
…					

774 # Changes to NEMA Standards Publications PS 3.15

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