DICOM Change Proposal

STATUS	Assigned
Date of Last Update	2025/10/28
Person Assigned	steven.nichols@gehealthcare.com
Submitter Name	Jeroen Medema, jeroen.medema@philips.com
Submission Date	2025/10/28

Change Number CP-2582
Log Summary: Improve the Transaction names of the Storage Commitment Service
Name of Standard PS3.18, PS3.2
Rationale for Change: Currently, the names of the Transactions are “Request” and “Result Check” respectively. “Request” is unclear and may be misleading, as there are also subsections named “Request”. “Result Check” likewise lacks an action-oriented name and does not clearly relate to the initiating transaction. HTTP semantics defined by the IETF describe operations as actions applied to a target resource (RFC 9110). In line with this model, transaction names in HTTP-based APIs commonly follow a verb–object pattern (e.g., “Check Commit Result”) rather than noun–noun constructions (e.g., “Commit Check”). This aligns with verb–object ordering already used throughout PS3.18 (e.g., Retrieve Study/Series/Instance, Search for Studies/Series/Instances, Store Instances, Create Workitem, Change Workitem State). Also, some naming and numbering is erroneous and requires correction.
Change Wording: See below. It is proposed to change “Request” to “Commit”, and “Result Check” to “Check Commit Result”, in line with the proposed Transaction names of the Send Transactions that are currently under discussion (Supplement 248): “Send” and “Check Send Result” respectively.

Update PS3.2, Annex N

…

N.5.3.6 Storage Commitment Web Service

This section provides details regarding the Storage Commitment Web Service. For an overview of supported Transactions and resources see Table N.1.3.5-1 Storage Commitment Service

N.5.3.6.1 Commit~~Request~~ Transaction - Storage Commitment Service

N.5.3.6.1.1 User Agent

The **Commit~~Request~~** Transaction user agent can request resources listed in Table N.5.3.6.1.1-1.

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

Table N.5.3.6.1.1-1. Resources for Commit~~Request~~ Transaction - User Agent

Resource	Comments
	See Resources path in Table 13.1.1-1 in PS3.18
~~C~~c*ommitment-requests*	

The **Commit~~Request~~** Transaction user agent supports Header Fields listed in Table N.5.3.6.1.1-2.

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

Table N.5.3.6.1.1-2. Header Fields for Commit~~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.5.3.6.1.2 Origin Server

The **Commit~~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.6.3.5.

The Request Transaction origin server supports resources listed in Table N.5.3.6.1.2-1.

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

Table N.5.3.6.1.2-1. Resources for Commit~~Request~~ Transaction - Origin Server

Resource	Comments
	See Resources path in Table 13.1.1-1 in PS3.18
commitment-requests	

The **Commit~~Request~~** Transaction origin server supports Header Fields listed in Table N.5.3.6.1.2-2.

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

Table N.5.3.6.1.2-2. Header Fields for Commit~~Request~~ Transaction - Origin Server

Header Field	Supported Values	Comments
Content-Type	*application /dicom+json* *application /dicom+xml*	

Header Field	Supported Values	Comments
	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.5.3.6.2 Check Commit Result ~~Check~~ Transaction - Storage Commitment Service

N.5.3.6.2.1 User Agent

The resources and header fields supported by the user agent for the **Check Commit** Result ~~**Check**~~ Transaction are the same as for the **Commit**~~**Request**~~ Transaction; see Section N.5.3.6.1.1.

N.5.3.6.2.2 Origin Server

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

The Base URI, resources, and header fields supported by the origin server for the Result Check Transaction are the same as for the **Commit**~~**Request**~~ Transaction; see Section N.5.3.6.1.2.

…

N.6.3.5 Storage Commitment Service Configuration

N.6.3.5.1 Request Transaction Configuration

Table N.6.3.5-1 lists configuration parameters for the **Commit**~~**Request**~~ Transaction of the Storage Commitment Service:

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

Table N.6.3.5-1. Commit~~Request~~ and Check Commit Result ~~Check~~ Transaction Parameters

Local Configuration Parameters - Commit~~Request~~ and Check Commit 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://<hostname>:<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*

Local Configuration Parameters - Commit~~Request~~ and Check Commit Result ~~Check~~ Transaction			
Parameter	**Configurable**	**Default Value**	**Comments**
			retrievable from the origin server.
<Specific Storage Commitment Service parameter>			
Remote Configuration Parameters - Commit~~Request~~ and Check Commit 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**	<emphasis role="bold">
	<<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.6.3.5.2 Check Commit Result ~~Check~~ Transaction Configuration

Table N.6.3.5-1 lists configuration parameters for the **Check Commit** Result ~~**Check**~~ Transaction of the Storage Commitment Service.

…

N.7.3.3.6 Storage Commitment Service

N.7.3.3.6.1 Commit~~Request~~ Transaction As Origin Server

Table N.7.3.3.6.1-1 lists the Status Codes that an origin server supports for the **Commit~~Request~~** Transaction of 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 **Commit~~Request~~** Transaction response as origin server.]*

Table N.7.3.3.6.1-1. Status Codes of Origin Server for Commit~~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*

N.7.3.3.6.2 Commit~~Request~~ Transaction As User Agent

Table N.7.3.3.6.2-1 lists the Status Codes that a user agent supports for the **Commit~~Request~~** 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 **Commit~~Request~~** Transaction response.]*

Table N.7.3.3.6.2-1. Status Codes of User Agent for Commit~~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*

N.7.3.3.6.3 Check Commit Result ~~Check~~ Transaction As Origin Server

Table N.7.3.3.6.3-1 lists the Status Codes that an origin server supports for the **Check Commit** Result ~~**Check**~~ Transaction of 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 **Check Commit** Result ~~**Check**~~ Transaction response as origin server.]*

Table N.7.3.3.6.3-1. Status Codes of Origin Server for Check Commit 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 **Check Commit Result**~~**result check**~~ request; this may be a temporary or permanent state*

N.7.3.3.6.4 Check Commit Result ~~Check~~ Transaction As User Agent

Table N.7.3.3.6.4-1 lists the Status Codes that a user agent supports for the **Check Commit** 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 **Check Commit** Result ~~**Check**~~ Transaction response.]*

Table N.7.3.3.6.4-1. Status Codes of User Agent for Check Commit 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*

Update PS3.18, Section 13

13 Storage Commitment Service and Resources

13.1 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 Section J “Storage Commitment Service Class” in 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 (ARR) pattern [Ekuan].

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 13.1-1 shows the possible scenarios of requesting storage commitment.

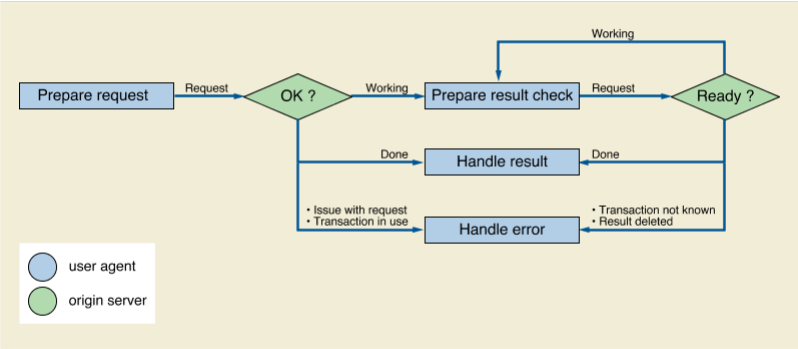

Figure 13.1-1. Process of the Storage Commitment Service

This starts when the user agent sends a **Commit~~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 **Commit~~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 **Commit~~Request~~** with Working, it behaves asynchronously, and is working on the request. In this case, the user agent needs to perform a **Check Commit** 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 **Check Commit** Result~~-Check~~ with Working, which will trigger the user agent to perform a **Check Commit** Result ~~Check~~ again. This process continues until the origin server responds with Done, finalizing the process.

For both the **Commit~~Request~~** and the **Check Commit** 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 Table 13.4.3-1 and Table 13.5.3-1 for more details.

…

13.2 Conformance

Implementations conforming to the Storage Commitment Service shall support the transactions listed as Required in Table 13.2-1.

Table 13.2-1. Required and Optional Transactions

Transaction	Support	Section
Commit~~Request~~	Required	Section 13.4
Check Commit Result~~-Check~~	Required	Section 13.5

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 committed 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 **Commit~~Request~~** will be available for the user agent.

13.3 Transactions Overview

The Storage Commitment Service consists of the transactions listed in Table 13.3-1.

Table 13.3-1. Storage Commitment Service Transactions

Transaction Name	Method	Payload		Description
		Request	Success Response	
Commit~~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.
Check Commit Result-**~~Check~~**	GET	N/A	Storage Commitment Result	Gets the result of a **Commit~~Request~~**.

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

13.4 Commit~~Request~~ Transaction

…

13.4.3.1 Status Codes

Table 13.4.3-1 shows some common status codes corresponding to this transaction. See also Section 8.5 for additional status codes.

Table 13.4.3-1. 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.

Status	Code	Meaning
	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 **Check Commit** Result ~~**Check**~~ transaction, as described in Section 13.5, 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 temporary or permanent state.

13.4.3.2 Response Header Fields

The origin server shall support Header Fields as required in Table 13.4.3-2.

Table 13.4.3-2. 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) **Check Commit Result**~~**result check**~~ or retrying the request.

…

13.5 Check Commit Result ~~Check~~ Transaction

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

Note

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

13.5.1 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
```

13.5.1.1 Target Resource

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

13.5.12.2 Query Parameters

The request has no Query Parameters.

13.5.12.3 Request Header Fields

The origin server shall support **Request~~Result Check~~** Header Fields as required in Table 13.5.1-2.

The user agent shall supply **Request~~Result Check~~** Header Fields as required in Table 13.5.1-2.

Note

The presence and values of the **Check Commit Result Request** Header Fields should be the same as those of the **Commit~~storage commitment~~** Request Header Fields.

Table 13.5.1-2. Request~~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.

...

13.5.3.1 Status Codes

Table 13.5.3-1 shows some common status codes corresponding to this transaction. See also Section 8.5 for additional status codes.

Table 13.5.3-1. Status Code Meaning

Status	Code	Meaning
Success	200 (OK)	The origin server finished processing the **Commit~~Request~~** transaction identified by the supplied Transaction UID (see Section 13.4); the payload contains the result.
	202 (Accepted)	The origin server has not yet finished processing the **Commit~~Request~~** transaction identified by the supplied Transaction UID; there is no payload. The user agent is expected to follow-up again with the **Check Commit** 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 **Check Commit** Result-**~~Check~~** request; this may be a temporary or permanent state.

Note

1. The 404 (Not Found) status code may be caused by an incorrect Transaction UID that has been supplied by the user agent, or the origin server may have deleted the applicable result.
2. The 410 (Gone) status code may be caused by the origin server deleting the applicable result, but still having a record of the Transaction UID.
3. When the 404 (Not Found) or the 410 (Gone) status code is returned, the user agent might initiate a new storage commitment request. When the 503 (Service Unavailable) status code is returned, the user agent might retry later with another **Check Commit** Result-**~~Check~~** transaction.

205 **...**

H Capabilities Description

…

Table H-1. Resources and Methods

Service	Resource	Transactions	Reference
…			
Storage Commitment requests (see Section 13.1.1)			
	commitment-requests	**Commit~~Request~~**	Section 13.4
		Check Commit Result ~~**Check**~~	Section 13.5
…			