

Fedora API Delta Document

This documentation is out of date and no longer maintained.

Please see: [Fedora API Spec and Delta Document Verification](#)

This page describes an effort to produce a “delta” document (or documents?) that characterize the differences between the API exposed by the current community Fedora implementation, and the [draft Fedora specification](#).

- [Specified Delta in Capabilities](#)
 - [CRUD / LDP](#)
 - [Versioning](#)
 - [Fixity](#)
 - [Authorization](#)
 - [Messaging](#)

Audience:

- Application/framework developers?
- Managers/stakeholders?
- Fedora committers?
- ?

Purpose:

- Allow the community to make decisions about aspects of fedora’s current API that are outside of any specification?
 - Breaking changes due to alignment with the specification (e.g. drop support for “snapshot” versioning, move to memento-based versioning in spec)
 - Deprecating features completely
 - Keeping features that are outside any specification, but allowable
- Allow the development of timelines for implementing specific breaking changes in the current Fedora impl?
- Aid in communicating “what’s changing” to the community?
- Aid in evaluating the proposed Fedora specification?
- Allow application developers to identify what they need to do (if anything) to update their applications so that they work with Fedora instances that follow the new specification?
- Allow application developers to identify which not-specified-but-allowable behaviors unique to the the current Fedora implementation their application relies on?
 - .. and either move away from them, or acknowledge and affirm them?

Outcomes:

- High-level overview document?
- Detailed/comprehensive section-by-section comparison with specs?

Specified Delta in Capabilities

CRUD / LDP

1. [External Binary Content](#)

- a. Spec: Must advertise support in the Accept-Post response header for each supported type parameter of supported Content-Type values (url and local-file)
- b. Spec: requests that would create or update a LDP-NR with a message/external-body with an unsupported access-type parameter must respond with HTTP 415 UNSUPPORTED MEDIA TYPE
 - i. For example, the following should return a 415:

```
curl -XPUT -H"Content-Type: message/external-body; access-type=ftp; NAME=\"some/file\"";  
site=\"example.com\" -i localhost:8080/rest/external
```

- c. Spec: Fedora servers receiving requests that would create or update a LDP-NR with a message/external-body must not accept the request if it cannot guarantee all of the response headers required by the LDP-NR interaction model in this specification.
 - i. Modeshape impl: Add support for Want-Digest on external content
- d. Spec: LDP-NR GET and HEAD responses should include a Content-Location header with a URI representation of the location of the external content if the Fedora server is proxying the content.
 - i. Modeshape impl: add support for Content-Location header on external content
- e. Spec: Requests that would create or update a LDP-NR with a message/external-body content type should respect the expiration parameter, if present, by copying content

- i. Modeshape impl: add support for copying content into the repository when "expiration" parameter is present in create/update requests
- f. Ensure 'Accept-Post' header includes 'message/external-body' url and local-file
- g. Respond with 415 code for other access-type values
- h. Ensure all LDP-NR headers are included in external content responses
- i. 'Location' header should include 'Content-Location' with URI of external content
- j. Add support for 'expiration' parameter, to copy content into the repo
 - First step could be 4xx response with 'constrainedby' Link

3.1.1

- a. NonRDFSsource link header must be supported, regardless of Content-Type

3.2.1

- a. Must allow interaction model to change to subtype of current model
- b. Must disallow PATCH to change ixn model to non-subtype of current model

Since the only types supported are NonRDFSsource, BasicContainer, DirectContainer and IndirectContainer (none of which are subtypes of any of the others), this effectively means that types cannot be changed after creation. In particular, DirectContainers and IndirectContainers cannot be created as BasicContainers and then PATCHed with the new type and membership predicates.

3.3

- a. POST creation must return a 'constrainedby' header indicating ixn model

3.3.1

- a. Ensure that LDP-NR creation requests fail with correct response code
 - 409 for mismatch
 - 400 for unsupported algorithm

3.4

- a. PUT update of resource must fail with 409 if changing ixn model to non-subtype

3.5.1

- a. MAY add support for 'PreferContainedDescriptions': not just URIs of contained

3.5.2

- a. Ensure correct behavior of returning 'Preference-Applied' header

3.5.3

- a. Add support for 'Want-Digest' header

3.6

- a. Ensure Digest header is the same of HEAD as GET
- b. Ensure payload headers are omitted on GET (see: rfc7231:section3.3)

3.7.1

- a. Add support for DEPTH header, values: 0 and infinity
- b. Respond with 400 for other values of DEPTH

Versioning

4.1.2.1

1. Accept 'Accept-Datetime' header on LDPRv's, on GET, to retrieve a memento from a particular resource at a particular time.

4.1.2.2

1. Include 'Link: rel="timemap"' on versioned resource pointing to LDPCv, on GET
2. Add response header Link: rel="timegate" referencing itself
3. Add response header Link: rel="original" referencing itself
4. If 'Accept-Datetime' header was included on the GET, include "Memento-Datetime" header to indicate response is a specific prior version
5. include 'Vary: Accept-Datetime' header in response

4.1.3.1

1. PUT on LDPRv creates a new LDPRm in the LDPCv, in a server managed versioning scenario

4.1.3.2

1. PUT request header to support 'Accept-Datetime', to indicate exactly what time a version should have associated with it.

4.1.3.3

1. Include 'Link: rel=timemap' pointing to LDPCv for versioned resources
 - On PUT response
 - Also include 'Vary: Accept-Datetime' header in response

4.2.1

1. Creation of versioned resource triggered by 'Link: rel=type' value=VersionedResource
 - An LDPRm is also created

4.2.3

1. TimeGate for LDPRm is LDPRv on GET.
2. Headers must include "Accept-Datetime" in GET request for the memento
3. Headers must include "Memento-Datetime" in response to GET request

4.2.5

1. Include following OPTIONS on LDPRm: GET, HEAD, OPTIONS, DELETE

4.2.6

1. Forbid PATCH on LDPRm

4.2.7

1. Forbid POST on LDPRm

4.2.8

1. Forbid PUT on LDPRm

4.3.1

1. Create LDPCv when versioned resource (LDPRv) is created
 - a. LDPCv is a TimeMap
2. Must respond to GET 'Accept: application/link-format'
3. LDPCv must not be contained by the LDPRv
4. If POST is supported, then any response from LDPCv must include 'Accept-Post: */*; p=0.0,' in header

4.3.2

1. Delete of LDPCv is allowed - mementos will also be deleted.
2. LDPR will remain intact, but the versioning interaction model will no longer appear
 - a. TimeGate / TimeMap links in Header will no longer appear
3. If POST is supported, then any response from LDPCv must include 'Accept-Post: */*; p=0.0,' in header

4.3.3

1. OPTIONS must show GET, HEAD, OPTIONS
2. OPTIONS may show DELETE, if supported
3. OPTIONS may show PATCH and POST if LDPCv itself has mutable properties
4. If POST is supported, then any response from LDPCv must include 'Accept-Post: */*; p=0.0,' in header

4.3.4

1. PATCH can not modify containment triples
2. If POST is supported, then any response from LDPCv must include 'Accept-Post: */*; p=0.0,' in header

4.3.5

1. POST to LDPCV should create new LDPRm for the LDPRv, contained by the LDPCv
2. If POST is supported, then any response from LDPCv must include 'Accept-Post: */*; p=0.0,' in header
3. POST with body should respond with 415 and constrainedby Link
4. LDPRm created on POST to LDPCv should respond with 'Location' header of LDPRm

? # "If an LDPCv does accept POST with request body..."? what would that mean?

 - a. Would this implicitly change the LDPRv?
5. Honor 'Memento-Datetime' header on LDPCv POST

4.4

1. 'Vary' header on LDPCv should be:
 - a. if POST is supported to create a memento: Vary-Post: Memento-Datetime,
 - b. if PUT/POST is supported on a LDPRv to create memento then add Vary-Post/Vary-Put: Memento-Datetime

Transmission Fixity

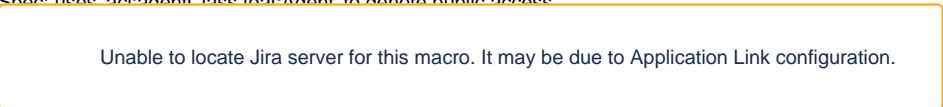
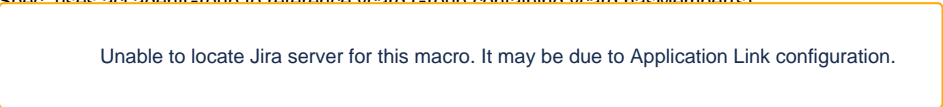
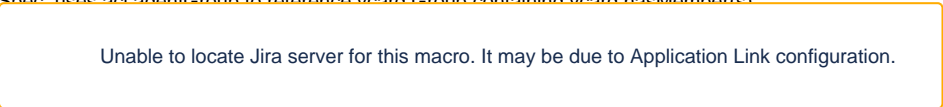
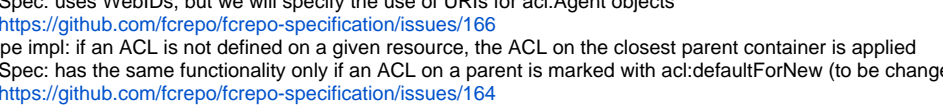
- **Summary:** No change.
- Modeshape impl: Add a Digest header to content uploads. The server compares the digest to the uploaded content's digest, and if they do not match, it will discard the content and return a 409 Conflict error status.
- Spec: Same

Persistence Fixity

- **Summary:** The exact mechanism has changed, but the basic approach is similar.
- Modeshape impl: Add "/fcr:fixity" to the end of a binary URL (advertised in the binary description with a triple with the predicate fedora:hasFixityService). The response RDF will contain one or more triples with the predicate premis:hasEventOutcome, and the object "SUCCESS" on success, and "BAD_CHECKSUM" and/or "BAD_SIZE" on failure.
- Spec: A HEAD or GET request to a binary with the Want-Digest header triggers a fixity check. The response will include a Digest header with the computed checksum. The client will need to compare the digest to the stored value.
- Support Want-Digest on HEAD and GET of LDP-NRs

Authorization

See: <https://github.com/fcrepo/fcrepo-specification/issues?q=is%3Aissue+is%3Aopen+label%3Atopic-authorization>

1. Modeshape impl: supports acl:Read and acl:Write
 - a. Spec: Also includes support for acl:Append and acl:Control
 - b. <https://github.com/fcrepo/fcrepo-specification/issues/170>
2. Modeshape impl: uses 'acl:agent foaf:Agent' to denote public access
 - a. Spec: uses 'acl:agentClass foaf:Agent' to denote public access
 - b. 
 - c. 
3. Modeshape impl: uses 'acl:agentClass foaf:Agent' to denote public access
 - a. Spec: uses 'acl:agentClass foaf:Agent' to denote public access
 - b. 
 - c. 
4. Modeshape impl: uses 'acl:agentClass foaf:Agent' to denote public access
 - a. Spec: uses 'acl:agentClass foaf:Agent' to denote public access
 - b. <https://github.com/fcrepo/fcrepo-specification/issues/166>
5. Modeshape impl: if an ACL is not defined on a given resource, the ACL on the closest parent container is applied
 - a. Spec: has the same functionality only if an ACL on a parent is marked with acl:defaultForNew (to be changed to acl:default)
 - b. <https://github.com/fcrepo/fcrepo-specification/issues/164>
6. Modeshape impl: uses 'acl:accessToClass' for targeting all resources with a given rdf:type value
 - a. Spec: Do not use acl:accessToClass, per the [SOLIDWAC](#) recommendation

Messaging

- **Summary:** The format for the messages will change to JSON-LD
- <https://github.com/fcrepo/fcrepo-adopters-guide/blob/master/docs/messaging.md>
- 6.1
 1. Emit message for synthetic triples added/removed
 2. Emit message for each resource deleted in a hierarchy
- 6.3
 - The notification serialization **must** conform to the [\[activitystreams-core\]](#) specification.
 - Wherever possible, data **should** be expressed using the [\[activitystreams-vocabulary\]](#).
 - Remove use of message headers (not a spec requirement to mentioned in [messaging.md](#))
 - Examples of messages:
 - minimal: <https://fcrepo.github.io/fcrepo-specification/#minimal-notification-example>
 - with additional information: <https://fcrepo.github.io/fcrepo-specification/#basic-notification-example>