

Versioning Delta/Specification Notes

Specification

- 4.1.3.1 In supporting PUT to LDPRv in order to create a LDPRm, is this just included for server managed versions? Or is PUT to an LDPRv intended to be addressable separately from a LDPR so as to trigger this behavior?
 - <https://github.com/fcrepo/fcrepo-specification/issues/215>
 - *Propose that PUT language is removed and that server managed behaviors are left up to implementors so long as they meet other requirements*
- 4.3.1 What is the default serialization for GET requests to LDPCv? The only required encoding is application/link-format.
 - This will depend on if the LDPCv remains the same object as the timemap
- 4.3.5 What is the expected behavior if a POST to an LDPCv is made which specifies a Memento-Datetime that is already in use by an existing LDPRm for that LDPRv?
 - Having two memento's with the same datetime would result in datetime negotiation yielding indeterminate results.
 - Ignore request and return a 412 response? Delete and replace the existing version?
 - This could be something we decide - perhaps the new one overwrites the existing one?
 - *Starting with a 412 response*
- 4.5 Does the modeshape fcrepo currently have a server managed option for versioning?
 - We need to make sure that a server managed scenario knows how to handle an LDPCv correctly - in that it doesn't version that resource if a user issues a PUT against it (as a user would if they want the original resource LDPRv to get versioned).

Delta

With Fedora API Specification:

- 4.1.1 Add HTTP header Link: rel="timegate" referencing itself
 - versions will need to be accessible via the TimeGate, which is the LDPRv. The timemap will still include the regular path to the LDPRm.
- 4.1.2.1 include 'Vary: Accept-Datetime' header in response
- 4.2.3 When requesting a Memento or a TimeGate, a Link header must be returned with type of "original" which points to the URI of the original. See 2.2.1. Link Relation Type "original"
- 4.2.3 "Memento-Datetime" should be returned when making a GET against a memento, indicating the timestamp of the memento
- 4.1.2.2 The TimeMap link may include "from" and "until" attributes, do we want to provide these? See 2.2.3. Link Relation Type "timemap"
- 4.1.2.1 How is fedora going to handle Accept-Datetime negotiation? nearest version? nearest version before requested time? Or is it even going to be used this way? See 3.1. Datetime Negotiation (first paragraph after example steps)
- 4.3.5 Are we going to continue allowing the "slug" header on requests to create versions?
- It would be helpful to record which Datetime Negotiation pattern the modeshape implementation uses in documentation

With Memento Specification:

- 2.2.1 - noted above in 4.2.3: on a GET/HEAD request to a LDPRm or TimeGate (the LDPRv, in this case) a Link header must be returned with relation type of 'original' which points to URI of the original resource (LDPRv)
- 2.2.2 - on a GET/HEAD request to a LDPR or LDPRm, a Link header must be returned with the relation type of 'timegate', which points to the the timegate for the resource.
 - multiple timegate Links are possible
- 2.2.3 - on a GET/HEAD request to a LDPR or LDPRm or TimeGate(which is the LDPR, in this case), a Link header must be returned with the relation type of 'timemap', which points to the timemap for the resource.
 - support "from" and "until" attributes? These would cause the returned representation of the time map to only include the interval requested.
 - the link should make use the 'type' attribute to indicate the mime type of the timemap.

LDPCv Example Response

```
# GET request to LDPCv
curl http://localhost/rest/a/fcr:versions -H "Accept: application/link-format" -v

< HTTP/1.1 200 OK
< Date: Thu, 21 Jun 2017 00:06:50 GMT
< Server: Apache
< Content-Type: application/link-format
< Allow: GET, HEAD, OPTIONS, POST, PATCH, DELETE
< Accept-Post: text/turtle,text/rdf+n3,text/n3,application/rdf+xml,application/n-triples,application/ld+json,
multipart/form-data,application/sparql-update
< Accept-Patch: application/sparql-update
< Connection: close

<http://localhost/rest/a>;rel="original timegate",
<http://localhost/rest/a/fcr:versions>
; rel="self";type="application/link-format"
; from="Tue, 20 Jun 2016 18:02:59 GMT"
; until="Wed, 09 Apr 2017 20:30:51 GMT",
<http://localhost/rest/a/fcr:versions/20160620180259356>; rel="memento";datetime="Tue, 20 Jun 2016 18:02:59
GMT",
<http://localhost/rest/a/fcr:versions/version_1>; rel="memento";datetime="Fri, 06 Jan 2017 18:50:11 GMT",
<http://localhost/rest/a/fcr:versions/20170409203051112>; rel="memento";datetime="Wed, 09 Apr 2017 20:30:51
GMT",
```

Example Interactions to Version a Resource or a Tree

```
# 1 - Create LDPRm of single resource
1) curl -XPOST /a/fcr:versions -H "Depth: 0"
   a) Create LDPRm /a/fcr:versions/20170919080311 in LDPCv
2) GET /a/fcr:versions/20170919080311
   </a/fcr:versions/20170919080311> ldp:contains </a/b>
3) GET /a
   </a> ldp:contains </a/b>

# 2 - Create LDPRm of tree
1) curl -XPOST /a/fcr:versions -H "Depth: infinity"
   a) Create LDPRm /a/fcr:versions/20170919080311
   b) Create LDPRm /a/b/fcr:versions/20170919080311
2) GET /a/fcr:versions/20170919080311
   </a/fcr:versions/20170919080311> ldp:contains </a/b>
3) GET /a
   </a> ldp:contains </a/b>
```

Note: if in example 2 the LDPRm were to have relationship:
 </a/fcr:versions/20170919080311> ldp:contains </a/b/fcr:versions/20170919080311>
 there would still be multiple parents containing

How does delete behave when multiple LDPRs (a LDPRv and a LDPRm) both ldp:contains the same child? What happens if you delete the LDPRm?

Handling Deletion of Referenced Resources / Mementos

Approach 1 - Prevent deletion

```
1) Given two resources, /a and /b, where:
</a> dc:related </b>

2) Create a version of /a with `POST /a/fcr:versions`, which creates LDPRm:
/a/fcr:versions/20170919162455

3) Delete /b
< HTTP/1.1 412 Precondition failed
< Date: Tue, 19 Sep 2017 19:55:15 GMT
< Content-Type: text/plain
< Content-Length: 77
< Server: Jetty(9.2.3.v20140905)
Cannot delete resource with incoming relationships from immutable resources.
```

Approach 2 - Allow deletion, ignore referential integrity

```
1) Given two resources, /a and /b, where:
</a> dc:related </b>

2) Create a version of /a with `POST /a/fcr:versions`, which creates LDPRm:
/a/fcr:versions/20170919162455

3) Delete /b
< HTTP/1.1 204

4) Get /a
</a> dc:related </b>

5) Get /b
< HTTP/1.1 410 Gone
```

Accept-Datetime Negotiation Example

```
Given a LDPRv with two mementos
* Rv => LDPRv, fedora:lastModified = 2017-09-13T20:11:31.291Z
* M1 => LDPRm, memento:datetime = 2017-09-08T18:01:33.535Z
* M0 => LDPRm, memento:datetime = 2017-09-04T09:20:12.030Z

1) GET Rv with no Accept-Datetime
Returns Rv as a LDPRv response.
2) GET Rv "Accept-Datetime: Thu, 1 Jan 2016 00:00:00 GMT" (prior to resource creation)
Response 404 not found
3) GET Rv "Accept-Datetime: Wed, 06 Sep 2017 00:00:00 GMT" (Between M0 and M1)
Returns M0 as a LDPRm response, with Content-Location referencing M0
4) GET Rv "Accept-Datetime: Sun, 10 Sep 2017 00:00:00 GMT" (Between M1 and Rv)
Return M1 as a LDPRm response, with Content-Location referencing M1
5) GET Rv "Accept-Datetime: Sat, 16 Sep 2017 00:00:00 GMT" (after Rv)
Returns Rv as a LDPRv response.
6) GET Rv "Accept-Datetime: Fri, 08 Sep 2017 18:01:33 GMT" (Exact match)
Return M1 as a LDPRm response, with Content-Location referencing M1
```

To summarize:

- 1) If no Accept-Datetime is provided, then the LDPRv is returned
- 2) If Accept-Datetime is provided, then the nearest LDPRm with a Memento-Datetime < than requested time, or LDPRv with created time < than requested time is returned.

Immutable LDPRms and Relaxing Referential Integrity

Fcrepo/Memento defines a past version of a LDPRm (aka a memento) as a fixed representation of the original LDPR at the time the version was created. However, a memento can be incorrectly modified in the following scenario, given memento M of original resource O, and another resource R:

1. M contains relation dc:related to R
2. Delete resource R
3. Modeshape automatically removes dc:related relation from M
4. M no longer matches O at the time that M was created.

This is because modeshape fedora maintains referential integrity - if a resource is deleted, all references to that resource are removed as well.

To address this, requirements for referential integrity would be relaxed for mementos such that they could continue to refer to the deleted resource. This means that mementos could have triples that reference objects that no longer exist.

This would come with some implications:

- When restoring a LDPRm, references would become active again, meaning that they could then be subject to removal. There would likely need to be error reporting to inform clients of this.
 - Need to determine what acceptable validation or outcomes would be for restoring when information could be lost.

Two options for how to handle this would include:

1. When an LDPRm is created from an LDPRv, all references to modeshape nodes would be converted to URIs. Since they would no longer be node references, they would not be cleaned up.
2. Instead of storing a LDPRm as a LDPRS, it could be stored as a RDF serialization of the LDPRS stored to a binary.

Versions will be of individual resources, versus tree snapshots

In Fedora 4, when a version of a resource is created, a snapshot of it and the entire tree of child resources it contains is captured. Moving ahead, the proposed design would instead only create a version of the requested resource, stopping there rather than versioning its child resources. The resource would be copied as it is and links to other resources would remain intact.

For example,

```
1) Create resources </a> and </a/b> as versionable resources (LDPRvs)
2) Create LDPRm of </a>
   Produces LDPRm </a/fcr:versions/20170919080311> in LDPCv
3) GET LDPRm </a/fcr:versions/20170919080311>, which returns:
   </a/fcr:versions/20170919080311> ldp:contains </a/b>
4) GET LDPRv /a
   </a> ldp:contains </a/b>
5) No LDPRm was created for </a/b>
```

Implications:

- Resource versions could now be managed individually, rather than creating unwanted trees, when the goal is only to version a single resource's metadata.
- Creating a snapshot of an entire tree would need to be triggered with many individual API requests.
- Retrieving or restoring a tree of resources to a particular point in time would require individual API requests with datetime negotiation.
- Creation and management of versions would shift from being a Modeshape-provided feature to the Fedora code base.

Draft Tickets

Add versioning response headers to LDPRv HEAD/GET responses -

The following headers should be added to responses to HEAD and GET requests to LDPRv's only when the resource is versioned (has the <http://fedora.info/definitions/fcrepo#VersionedResource> type):

- Vary: accept-datetime
- Include 'Link: rel="timagate"' referencing itself.
- Include 'Link: rel="timemap"' on versioned resource pointing to LDPCv.
- Since the LDPRv is also its own timemap, it must return an 'original' link header referencing itself

See Pattern 1.2 in <https://tools.ietf.org/html/rfc7090#page-19> for details

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Support Datetime negotiation for LDPRvs

Implement support for datetime negotiation for the retrieval of previous versions of a resource as follows:

- GET requests to LDPRv URIs must support the 'Accept-Datetime' header
- It must follow the syntax outlined in <https://tools.ietf.org/html/rfc7089#section-2.1.1>
- When a 'Accept-Datetime' is provided:
 - Must respond with a 302 status code and an empty body
 - Response must contain 'Location' header containing the URI of the LDPRm or LDPRv that was negotiated
 - Must contain all other standard LDPRm GET response headers
- When a datetime is provided, Fedora will respond with the past LDPRm chronologically nearest (min-past) to the given header as follows:
 - See: [Datetime negotiation example](#)

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Add versioning headers for LDPRm HEAD/GET responses

Responses to LDPRm HEAD and GET requests must include headers as outlined below.

- Add HTTP header Link: rel="timagate" to the LDPRv
- Include 'Link: rel="timemap"' on versioned resource pointing to LDPCv
- add 'original' link header referencing LDPRv
- Add 'Memento-Datetime' header

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Support OPTIONS request on LDPRms

Response to include header 'Allow: GET, HEAD, OPTIONS, DELETE'

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Prevent modification requests to LDPRm

PATCH/POST/PUT requests should all return 405 responses

DELETE - If a resource is deleted from the repository, it appears as if all triples across the repository that have that resource as an object will get removed. Even in a versioned resource. According to the API spec, mementos need to be [immutable](#). This particular behavior of cleaning up will have to [change where mementos are consumed](#).

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

PUT requests to LDPRv (work in progress)

See <https://fcrepo.github.io/fcrepo-specification/#httpput>

If a Accept-Datetime is provided on a regular PUT request to an LDPRv

- A new LDPRm is created from the LDPRv
- The LDPRm will have a Memento-Datetime matching the provided Accept-Datetime
- Response includes 'Link: rel="timemap"' pointing to LDPCv for versioned resources
- include 'Vary: Accept-Datetime' header in response
- If the LDPCv for the LDPRv does not already exist, then it will be created and contain the new LDPRm

POST request to LDPCv to create LDPRm

Update POST to LDPCv to include support for versioning headers. In practice, the LDPCv for object with uri <<http://localhost/rest/a>> would be <<http://localhost/rest/a/fcr:versions>>

See: <https://fcrepo.github.io/fcrepo-specification/#dpcvpost>

- if the request contained a body, then a new LDPRm is created from the submitted body
- if the request body was empty, then a new LDPRm is created from the current version of the LDPRv.
- If a Memento-Datetime header is provided as part of this POST request, then the newly created LDPRm must attempt to use it as the memento datetime
 - If a LDPRm already exists with the same Memento-Datetime specified, then the LDPRm is not created and a '412 already exists' response is returned.
- Response from a successful creation should return a LDPRm headers, particularly the Memento-Datetime of the new LDPRm.
- Response must include a Location header referencing the URL of the newly created LDPRm.

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Remove Support for Slug Header on POST to LDPCv

- Currently the modeshape fedora will request a "Slug" header be supplied to use as the label of the version. This should be taken out, to be replaced by the Memento-Datetime header, if that's present. The label should be a timestamp per the [Memento DateTime Section](#) of the Memento spec.

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Delete a LDPCv

See: <https://fcrepo.github.io/fcrepo-specification/#ldpcvdelete>

- Deletes the LDPCv, which has the affect of also deleting all of the LDPRm's contained by it.
- The versioning interaction model is removed from the LDPRv (<http://fedora.info/definitions/fcrepo#VersionedResource>).

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Delete a LDPRv

(this one might not actually be a ticket, since the behavior, though not totally correct, is in line with what we're targeting.)

DELETE of a LDPR is part of the fedora implementation currently, and will stay part of it. Per discussion on the fedora-tech list, linking the lifecycle of a LDPR and it's corresponding LDPRm's is okay. When the LDPRv is deleted, the mementos will also be deleted.

There is the concern about deleting LDPRs and LDPRm's that are referenced by other resources or versions of resources.

Implement application/link-format for TimeMaps

See: <https://fcrepo.github.io/fcrepo-specification/#general-3>

Requests to an LDPCv/TimeMap for a LDPRv must support a "application/link-format" response type, as shown in the example <https://tools.ietf.org/html/rfc7089#page-37>

- This response type is NOT required to include all statements in the LDPCv's graph, only those that are required by the link-format, outlined <https://tools.ietf.org/html/rfc7089#page-36>
- Including: URL of LDPRv for this timemap, all LDPRm's in this LDPCv with their memento date times, timegate link, link to the LDPCv

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

HEAD / GET on LDPCv

- Must indicate [TimeMap](#) in the same way it indicates the Container interaction model of the resource via HTTP headers. See: <https://fcrepo.github.io/fcrepo-specification/#general-3>
- Must return Vary-Post and Vary-Put with value 'Memento-Datetime'

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Enable versioning on a new LDPR

A PUT or POST request to create an object will make a resource versionable if it includes header **Link: rel="type"** with type of <http://fedora.info/definitions/fcrepo#VersionedResource>

1. A LDPR will be created as a LDPRv with the versioning type.
2. A LDPCv will be created
3. A LDPRm will be generated, contained by the LDPCv.

The newly created LDPCv:

- must not be a child of the LDPRv
- must be a LDPC (<https://fcrepo.github.io/fcrepo-specification/#LDPC>)
- must be track its relationship to the LDPRv
- Note: it will be replacing the current "fcr:versions" path, which was previously used to access modeshape versions. Versioning will no longer be using built-in modeshape tree snapshotting features

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Enable Versioning on an Existing LDPR

Note: this is still under discussion since an empty PUT to an LDPR may not be valid.

A PUT request to an Existing LDPR will make a resource versionable if it includes header **Link: rel="type"** with type of <http://fedora.info/definitions/fcrepo#VersionedResource>

1. The versioning type will be added to the LDPR, making it a LDPRv.

2. A LDPCv will be created for the LDPRv
3. A LDPRm will be generated, contained by the LDPCv

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Restore a LDPRm

Discussion of the details for this feature are still ongoing at <https://github.com/fcrepo/fcrepo-specification/issues/217>

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Answered questions

- 4.1.1 Should the "timegate" link only be present when an object is versioned or should all LDPR that could be versioned provide this link?
 - Yes, it should always be present
 - This seems to imply that all resources are LDPRv's from the start
- ~~4.3.1 Why does each memento need a separate TimeMap to be created, rather than sharing one for all of the LDPRv and LDPRm of one LDPR?~~
 - This was a misinterpretation. In the modeshape implementation, currently the timemap is the fcr:versions endpoint.
- ~~4.3.1 LDPCv must not be contained by the LDPRv~~
 - ~~does that mean that fcr:versions will no longer be a subpath of the LDPR?~~
 - This only applies to ldp:contains, the uri is irrelevant
- 4.1.2 The Fcrepo spec doesn't mention it directly, but is there a conflict between the Memento "Content-Location" header and External File's "Content-Location"? Can they both appear in the same response?
 - External File Content-Location header just appears on LDP-NR HEAD/GET requests, while Memento uses the header for responses from LDPRm and maybe LDPRv, both of which are LDP-RS's. So the headers should not both appear on the same response
- 4.3.1 How do you determine what serializations are available for TimeMaps? HEAD is not currently supported and is not in the Memento spec.
 - GET response formats are not actually listed in any other requests, only for Post and Patch.
 - However, it would likely be helpful to include info about HEAD and GET requests in the spec
 - <https://github.com/fcrepo/fcrepo-specification/issues/223>