Fedora API Test Suite Summary

for { implementation name: please use --implementation-name to specify } { implementation version: please use --implementation-version to specify}

Req Level	Num Pass	Num Fail	Num Skip	% Pass
MUST	137	6	10	96%
SHOULD	38	5	0	88%
MAY	28	1	13	97%
Total	203	12	23	94%

Specification Section	Req Level	Result	Test Description
3.1.1-A-1	MUST	PASS	Implementations must support the creation and management of [LDP] Containers.
3.1.1-A-2	MAY	PASS	Implementations may support the creation and management of [LDP] Direct Containers
3.1.1-A-3	MAY	PASS	Implementations may support the creation and management of [LDP] Indirect Containers
3.1.1-B	MUST	PASS	LDP Containers must distinguish [containment triples]
3.1.1-C	MUST	PASS	LDP Containers must distinguish [membership] triples.
3.1.1-D	MUST	PASS	LDP Containers must distinguish [minimal-container] triples.
3.1.2-A	MUST	PASS	Implementations MUST allow the membership constant URI to be set via the ldp:membershipResource property of the content RDF on container creation.
3.1.2-B	MUST	FAIL	Implementations MUST set the ldp:membershipResource by default when not specified on creation.
3.1.2-C	SHOULD	FAIL	Implementations SHOULD set the ldp:membershipResource to the LDPC by default when not specified on creation.
3.1.2-D	MAY	PASS	Implementations may allow the membership constant URI to be updated by subsequent PUT requests that change the ldp:membershipResource property of the resource content.
3.1.2-E	MAY	PASS	Implementations may allow the membership constant URI to be updated by subsequent PATCH requests that change the ldp:membershipResource property of the resource content.

3.1.2-G-1	MUST	PASS	Implementations must allow the membership predicate to be set via either the ldp:hasMemberRelation or ldp:isMemberOfRelation property of the content RDF on container creation, or otherwise default to an implementation defined value. Implementations should use the default <> ldp:hasMemberRelation ldp:member
3.1.2-G-2	MUST	PASS	Implementations must allow the membership predicate to be set via ldp:isMemberOfRelation property of the content RDF on container creation, or otherwise default to an implementation defined value. Implementations should use the default <> ldp:hasMemberRelation ldp:member
3.1.2-H	MUST	FAIL	Implementations must allow the membership predicate to be set by default to an implementation defined value.
3.1.2-I	SHOULD	FAIL	Implementations should use the default <> ldp:hasMemberRelation ldp:member
3.1.2-J	MAY	PASS	Implementations may allow the membership predicate to be updated by subsequent PUT requests that change the ldp:hasMemberRelation property of the resource content.
3.1.2-K	MAY	PASS	Implementations may allow the membership predicate to be updated by subsequent PATCH requests that change the ldp:hasMemberRelation property of the resource content.
3.1.2-L	MAY	PASS	Implementations may allow the membership predicate to be updated by subsequent PUT requests that change the ldp:isMemberOfRelation property of the resource content.
3.1.2-M	MAY	PASS	Implementations may allow the membership predicate to be updated by subsequent PATCH requests that change the ldp:isMemberOfRelation property of the resource content.
3.1.3-A	MUST	PASS	Implementations MUST allow the indirect container's membership constant URI to be set via the ldp:membershipResource property of the content RDF on container creation.
3.1.3-B	MUST	FAIL	Implementations MUST set the indirect container's ldp:membershipResource by default when not specified on creation.
3.1.3-C	SHOULD	FAIL	Implementations SHOULD set the indirect container's ldp:membershipResource to the LDPC by default when not specified on creation.
3.1.3-D	MAY	PASS	Implementations may allow the indirect container's membership constant URI to be updated by subsequent PUT

			requests that change the ldp:membershipResource property of the resource content.
3.1.3-E	MAY	PASS	Implementations may allow the indirect container's membership constant URI to be updated by subsequent PATCH requests that change the ldp:membershipResource property of the resource content.
3.1.3-F	MUST	PASS	Implementations must allow the membership predicate to be set on indirect containers via either the ldp:hasMemberRelation or ldp:isMemberOfRelation property of the content RDF on container creation.
3.1.3-G	MUST	PASS	Implementations must allow the membership predicate to be set on indirect containersvia ldp:isMemberOfRelation property of the content RDF on container creation.
3.1.3-H	MUST	FAIL	Implementations must allow the indirect container's membership predicate to be set by default to an implementation defined value.
3.1.3-I	SHOULD	FAIL	Implementations should use the default <> ldp:hasMemberRelation ldp:member
3.1.3-J	MAY	PASS	Implementations may allow the membership predicate to be updated by subsequent PUT requests that change the ldp:hasMemberRelation property of the resource content.
3.1.3-K	MAY	PASS	Implementations may allow the membership predicate to be updated by subsequent PATCH requests that change the ldp:hasMemberRelation property of the resource content.
3.1.3-L	MAY	PASS	Implementations may allow the membership predicate to be updated by subsequent PUT requests that change the ldp:isMemberOfRelation property of the resource content.
3.1.3-M	MAY	PASS	Implementations may allow the membership predicate to be updated by subsequent PATCH requests that change the ldp:isMemberOfRelation property of the resource content.
3.1.3-N	MUST	PASS	Implementations must allow the ldp:insertedContentRelation property to be set via the content RDF on container creation
3.1.3-O	MUST	FAIL	Implementations must allow the ldp:insertedContentRelation property to be set by default to an implementation defined value.
3.1.3-P	SHOULD	FAIL	Implementations SHOULD allow the ldp:insertedContentRelation property to be set by default to ldp:MemberSubject.
3.1.3-Q	MAY	PASS	Implementations may allow the ldp:insertedContentRelation

			property to be updated via the content RDF by subsequent PUT requests.
3.1.3-R	MAY	PASS	Implementations may allow the ldp:insertedContentRelation property to be updated via the content RDF by subsequent PATCH requests.
3.1.4-A	SHOULD	PASS	If, in a successful resource creation request, a Link: rel="type" request header specifies the LDP-NR interaction model (http://www.w3.org/ns/ldp#NonRDFSource, regardless of Content-Type: value), then the server should handle subsequent requests to the newly created resource as if it is an LDP-NR. ([LDP] 5.2.3.4 extension)
3.1.4-B	SHOULD	PASS	If, in a successful resource creation request, a Link: rel="type" request header specifies the LDP-NR interaction model (http://www.w3.org/ns/ldp#NonRDFSource, regardless of Content-Type: value), then the server should handle subsequent requests to the newly created resource as if it is an LDP-NR. ([LDP] 5.2.3.4 extension)
3.10.1-A	MAY	SKIPPED	Implementations may include the X-State-Token header field in GET responses to provide a token representing the current state of resource. If provided, this value must change whenever the underlying state of the resource has changed.
3.10.1-B	MAY	SKIPPED	Implementations may include the X-State-Token header field in HEAD responses to provide a token representing the current state of resource. If provided, this value must change whenever the underlying state of the resource has changed.
3.10.2-A	MUST	SKIPPED	A client may include the X-If-State-Token header field in a PATCH request to make the request conditional on the resource's current state token matching the client's value.
3.10.2-B	MAY	PASS	A client may include the X-If-State-Token header field in a PATCH request to make the request conditional on the resource's current state token matching the client's value. If an implementation does not support state tokens, it may ignore any X-If-State-Token header in HTTP PATCH requests.
3.10.2-C	MUST	SKIPPED	A client may include the X-If-State-Token header field in a PATCH request to make the request conditional on the resource's current state token matching the client's value. An HTTP PATCH request that includes an X-If-State-Token header must be rejected with a 412 (Precondition Failed) response if the implementation supports state tokens, but the client-supplied value does not match the resource's current state token.
3.10.2-D	MUST	SKIPPED	A client may include the X-If-State-Token header field in a PUT request to make the request conditional on the resource's

			current state token matching the client's value.
3.10.2-E	MAY	PASS	A client may include the X-If-State-Token header field in a PUT request to make the request conditional on the resource's current state token matching the client's value. If an implementation does not support state tokens, it may ignore any X-If-State-Token header in HTTP PUT requests.
3.10.2-F	MUST	SKIPPED	A client may include the X-If-State-Token header field in a PUT request to make the request conditional on the resource's current state token matching the client's value. An HTTP PUT request that includes an X-If-State-Token header must be rejected with a 412 (Precondition Failed) response if the implementation supports state tokens, but the client-supplied value does not match the resource's current state token.
3.2.1-A	SHOULD	PASS	In addition to the requirements of [LDP], an implementation ... should support the value http://fedora.info/definitions/fcrepo#PreferInboundReferences for the Prefer header when making GET requests on LDPC resources.
3.2.1-B	MAY	PASS	In addition to the requirements of [LDP], an implementation ... may support the value http://www.w3.org/ns/oa#PreferContainedDescriptions for the Prefer header when making GET requests on LDPC resources.
3.2.2-A	MUST	PASS	Responses to GET requests that apply a Prefer request header to any LDP-RS must include the Preference-Applied response header as defined in [RFC7240] section 3.
3.2.2-B	MUST	PASS	When a GET request is made to an LDP-RS that describes an associated LDP-NR (3.5 HTTP POST and [LDP]5.2.3.12),the response must include a Link: rel="describes" header referencing the LDP-NR in question, as defined in [RFC6892].
3.2.3-A-1	MUST	PASS	Testing for supported digest: md5 . GET requests to any LDP-NR must correctly respond to the Want-Digest header defined in [RFC3230]
3.2.3-A-2	MUST	PASS	Testing for supported digest: sha . GET requests to any LDP-NR must correctly respond to the Want-Digest header defined in [RFC3230]
3.2.3-A-3	MUST	PASS	Testing for supported digest: sha-256 . GET requests to any LDP-NR must correctly respond to the Want-Digest header defined in [RFC3230]
3.2.3-B	MUST	PASS	Testing for two supported digests with no weights GET requests to any LDP-NR must correctly respond to the Want-

			Digest header defined in [RFC3230]
3.2.3-C	MUST	PASS	Testing for two supported digests with different weightsGET requests to any LDP-NR must correctly respond to the Want-Digest header defined in [RFC3230]
3.2.3-D	MUST	PASS	Testing for two supported digests with different weights q=0.3,q=0 GET requests to any LDP-NR must correctly respond to the Want-Digest header defined in [RFC3230]
3.2.3-E	MUST	PASS	Testing for one supported digest and one unsupported digest.GET requests to any LDP-NR must correctly respond to the Want-Digest header defined in [RFC3230]
3.2.3-F	MUST	PASS	Testing that unsupported digest is rejected with a 400.GET requests to any LDP-NR must correctly respond to the Want-Digest header defined in [RFC3230].
3.3-A	MUST	PASS	The HEAD method is identical to GET except that the server must not return a message-body in the response, as specified in [RFC7231] section 4.3.2.
3.3-B	MUST	PASS	The server must send the same Digest header in the response as it would have sent if the request had been a GET (or omit it if it would have been omitted for a GET).
3.3-C	SHOULD	PASS	In other cases, The server should send the same headers in response to a HEAD request as it would have sent if the request had been a GET, except that the payload headers (defined in [RFC7231] section 3.3) may be omitted.
3.4-A	MUST	PASS	Any LDPR must support OPTIONS per [LDP] 4.2.8. 4.2.8.1 LDP servers must support the HTTP OPTIONS method.
3.4-B	MUST	PASS	Any LDPR must support OPTIONS per [LDP] 4.2.8. LDP servers must indicate their support for HTTP Methods by responding to a HTTP OPTIONS request on the LDPR's URL with the HTTP Method tokens in the HTTP response header Allow.
3.5-A	MUST	PASS	Any LDPC (except Version Containers (LDPCv)) must support POST ([LDP] 4.2.3 / 5.2.3).
3.5.1-A	MUST	PASS	Any LDPC must support creation of LDP-NRs on POST ([LDP] 5.2.3.3 may becomes must).
3.5.1-B	MUST	PASS	On creation of an LDP-NR, an implementation must create an associated LDP-RS describing that LDP-NR ([LDP] 5.2.3.12 may becomes must).
3.5.1-C	MUST	PASS	An HTTP POST request that would create an LDP-NR and includes a Digest header (as described in [RFC3230]) for

			which the instance-digest in that header does not match that of the new LDP-NR must be rejected with a 409 Conflict response.
3.5.1-D	SHOULD	PASS	An HTTP POST request that includes an unsupported Digest type (as described in [RFC3230]), should be rejected with a 400 Bad Request response.
3.6-A	MAY	FAIL	Implementations MAY allow the interaction model of an existing resource to be changed by specification of a new LDP type in a rel="type" link in the HTTP Link header
3.6-B	MUST	FAIL	When accepting a PUT request against an existant resource, an HTTP Link: rel="type" header may be included. If that type is a value in the LDP namespace and is not either a current type of the resource or a subtype of a current type of the resource, the request must be rejected with a 409 Conflict response.
3.6.1-A	MUST	PASS	Any LDP-RS must support PUT to update statements that are not server-managed triples (as defined in [LDP] 2). [LDP] 4.2.4.1 and 4.2.4.3 remain in effect.
3.6.1-B	MUST	PASS	If an otherwise valid HTTP PUT request is received that attempts to modify resource statements that a server disallows (not ignores per [LDP] 4.2.4.1), the server must fail the request by responding with a 4xx range status code (e.g. 409 Conflict).
3.6.1-C	MUST	PASS	The server must provide a corresponding response body containing information about which statements could not be persisted. ([LDP] 4.2.4.4 should becomes must).
3.6.1-D	MUST	PASS	In that response, the restrictions causing such a request to fail must be described in a resource indicated by a Link: rel="http://www.w3.org/ns/ldp#constrainedBy" response header per [LDP] 4.2.1.6.
3.6.2-A	MUST	PASS	Any LDP-NR must support PUT to replace the binary content of that resource.
3.6.2-B	MUST	PASS	An HTTP PUT request that includes a Digest header (as described in [RFC3230]) for which any instance-digest in that header does not match the instance it describes, must be rejected with a 409 Conflict response.
3.6.2-C	SHOULD	PASS	An HTTP PUT request that includes an unsupported Digest type (as described in [RFC3230]), should be rejected with a 400 (Bad Request) response.
3.6.3-A	MAY	PASS	Implementations may accept HTTP PUT to create resources

3.6.3-B	MAY	PASS	Implementations may accept HTTP PUT to create non-RDF resources
3.6.3-C	MUST	PASS	On creation of an LDP-NR with HTTP PUT, implementations MUST create an associated LDP-RS describing that LDP-NR in the same way that it would when 3.5.1 Creating LDP-NRs with HTTP POST
3.7-A	MUST	PASS	Any LDP-RS must support PATCH ([LDP] 4.2.7 may becomes must). [sparql11-update] must be an accepted content-type for PATCH.
3.7-B	MAY	SKIPPED	Other content-types (e.g. [ldpatch]) may be available.
3.7-C	MUST	PASS	If an otherwise valid HTTP PATCH request is received that attempts to modify statements to a resource that a server disallows (not ignores per [LDP] 4.2.4.1), the server must fail the request by responding with a 4xx range status code (e.g. 409 Conflict).
3.7-D	MUST	PASS	The server must provide a corresponding response body containing information about which statements could not be persisted. ([LDP] 4.2.4.4 should becomes must).
3.7-E	MUST	PASS	In that response, the restrictions causing such a request to fail must be described in a resource indicated by a Link: rel="http://www.w3.org/ns/ldp#constrainedBy" response header per [LDP] 4.2.1.6.
3.7-F	MUST	PASS	A successful PATCH request must respond with a 2xx status code; the specific code in the 2xx range may vary according to the response body or request state.
3.7.1	MUST	PASS	The server should not allow HTTP PATCH to update an LDPC's containment triples; if the server receives such a request, it should respond with a 409 (Conflict) status code.
3.7.2	MUST	PASS	The server must disallow a PATCH request that would change the LDP interaction model of a resource to a type that is not a subtype of the current resource type. That request must be rejected with a 409 Conflict response.
3.8.1-A	SHOULD	PASS	An implementation that cannot recurse should not advertise DELETE in response to OPTIONS requests for containers with contained resources.
3.8.1-C	MUST	PASS	An implementation must not return a 200 (OK) or 204 (No Content) response unless the entire operation successfully completed.
3.8.1-D	MUST	PASS	An implementation must not emit a message that implies the successful DELETE of a resource until the resource has been

			successfully removed.
3.9-A-1	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-A-1b	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-A-2	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-A-2b	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-A-3	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-A-3b	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-B-1	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-B-1b	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-B-2	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-B-2b	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"

3.9-B-3	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-B-3b	SHOULD	PASS	Fedora servers should support the creation of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-C-1	SHOULD	PASS	Fedora servers should support the update of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-C-1b	SHOULD	PASS	Fedora servers should support the update of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-C-2	SHOULD	PASS	Fedora servers should support the update of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-C-2b	SHOULD	PASS	Fedora servers should support the update of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-C-3	SHOULD	PASS	Fedora servers should support the update of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-C-3b	SHOULD	PASS	Fedora servers should support the update of LDP-NRs with content external to the request entity, as indicated by a link with rel="http://fedora.info/definitions/fcrepo#ExternalContent"
3.9-D-1	MUST	SKIPPED	Fedora servers that do not support the creation of LDP-NRs with content external must reject with a 4xx range status code
3.9-D-2	MUST	SKIPPED	Fedora servers that do not support the creation of LDP-NRs with content external must describe this restriction in a resource indicated by a rel="http://www.w3.org/ns/ldp#constrainedBy" link in the Link response header.
3.9-E-1	MUST	PASS	Fedora servers must use the handling attribute in the external content link to determine how to process the request. At least

			one of the following handling attributes must be supported: copy, redirect, and/or proxy.
3.9-E-2	MUST	PASS	Fedora servers must reject with a 4xx range status code requests for which the handling attribute is not present or cannot be respected.
3.9-E-3	MUST	PASS	In the case that the specified handling cannot be respected, the restrictions causing the request to fail must be described in a resource indicated by a rel="http://www.w3.org/ns/ldp#constrainedBy" link in the Link response header.
3.9-F-1	MUST	PASS	Fedora servers must use the value of the type attribute in the external content link as the media type of the external content, if provided.
3.9-F-2	MAY	PASS	Fedora servers must use the value of the type attribute in the external content link as the media type of the external content, if provided. If there is no type attribute: Servers may use the media type obtained when accessing the external content via the specified scheme (e.g. the Content-Type header for external content accessed via http).
3.9-F-3	MAY	PASS	Fedora servers must use the value of the type attribute in the external content link as the media type of the external content, if provided. If there is no type attribute: Servers may use a default media type.
3.9-F-4	MAY	PASS	Fedora servers must use the value of the type attribute in the external content link as the media type of the external content, if provided. If there is no type attribute: Servers may reject the request with a 4xx range status code.
3.9-F-5	SHOULD	PASS	Fedora servers must use the value of the type attribute in the external content link as the media type of the external content, if provided. Any Content-Type header in the request should be ignored.
3.9-F-6	SHOULD	PASS	Fedora servers must use the value of the type attribute in the external content link as the media type of the external content, if provided. Any Content-Type header in the request should be ignored.
3.9-G-1	MUST	PASS	A Fedora server receiving requests that would create or update an LDP-NR with content external to the request entity must reject request if it cannot guarantee all of the response headers required by the LDP-NR interaction model in this specification.
3.9.1	MUST	PASS	Fedora servers supporting external content MUST include "Accept-External-Content-Handling" header in response to

			"OPTIONS" request.
3.9.3-A-1	MUST	PASS	Fedora servers supporting "redirect" external content types MUST correctly respond to the "Want-Digest" header.
3.9.3-A-2	MUST	PASS	Fedora servers supporting "redirect" external content types MUST correctly respond to the "Want-Digest" header.
3.9.3-B-1	MUST	PASS	A successful response to a GET request for external content with handling of redirect must have status code of either 302 (Found) or 307 (Temporary Redirect)
3.9.3-B-2	MUST	PASS	A successful response to a HEAD request for external content with handling of redirect must have status code of either 302 (Found) or 307 (Temporary Redirect)
4-A	MAY	SKIPPED	Implementations may allow a subsequent PUT request with a rel="type" link in the Link header specifying type http://mementoweb.org/ns#OriginalResource to convert an existing LDPR into an LDPRv. If such a conversion from an LDPR to an LDPRv is supported, it must be accompanied by the creation of a version container (LDPCv), as noted above.
4.0-A	MUST	PASS	When an LDPR is created with a rel="type" link in the Link header specifying type http://mementoweb.org/ns#OriginalResource to indicate versioning, it MUST be created as an LDPRv and a version container (LDPCv) MUST be created to contain Memento resources
4.0-B	MAY	SKIPPED	Implementations MAY allow a subsequent PUT request with a rel="type" link in the Link header specifying type http://mementoweb.org/ns#OriginalResource to convert an existing LDPR into an LDPRv. If such a conversion from an LDPR to an LDPRv is supported, it MUST be accompanied by the creation of a version container (LDPCv), as noted above.
4.1.1-A-1	SHOULD	PASS	If no LDPRm is appropriate to the Accept-Datetime value, implementations should return a 406 (Unacceptable).
4.1.1-A-2	MUST	PASS	The Accept-Datetime header is used to request a past state, exactly as per [RFC7089] section 2.1.1. A successful response must be a 302 (Found) redirect to the appropriate LDPRm.
4.1.1-B	MUST	PASS	The response to a GET request on an LDPRv must return a rel="timegate" Link header referencing itself
4.1.1-C	MUST	PASS	The response to a GET request on an LDPRv must return a rel="timegate" Link header referencing itself

4.1.1-D	MUST	PASS	The response to a GET request on an LDPRv must return a <http://mementoweb.org/ns#OriginalResource>; rel="type" link in the Link header.
4.1.1-E	MUST	PASS	The response to a GET request on an LDPRv must return a <http://mementoweb.org/ns#OriginalResource>; rel="type" link in the Link header.
4.1.1-F	MUST	PASS	The response to a GET request on an LDPRv must return At least one rel="timemap" link in the Link header referencing an associated LDPCv
4.1.1-G	MUST	PASS	The response to a GET request on an LDPRv must return a Vary: Accept-Datetime header, exactly as per [RFC7089] section 2.1.2.
4.1.2-A	MUST	PASS	Must support PUT for creating new LDPRv
4.1.2-B	MUST	PASS	Must support PUT for updating existing LDPRvs
4.1.2-C	MUST	PASS	Must support PUT for creating new LDPNRv
4.1.2-D	MUST	PASS	Must support PUT for updating existing LDPNRvs
4.2.1-A	MUST	PASS	LDPR mementos must support GET
4.2.1-B	MUST	PASS	LDP-NR mementos must support GET
4.2.1-C	MUST	PASS	TimeGate for an LDP-RS memento is the original versioned LDP-RS
4.2.1-D	MUST	PASS	TimeGate for an LDP-NR memento is the original versioned LDP-NR
4.2.1-E	MUST	PASS	Any response to a GET request on an LDP-RS Memento must include a <http://mementoweb.org/ns#Memento>; rel="type" link in the Link header
4.2.1-F	MUST	PASS	Any response to a GET request on an LDP-NR Memento must include a <http://mementoweb.org/ns#Memento>; rel="type" link in the Link header
4.2.2-A	MUST	PASS	LDPRm resources must support OPTIONS
4.2.2-B	MUST	PASS	A response to an OPTIONS request must include Allow: GET, HEAD, OPTIONS
4.2.2-C	MAY	PASS	A response to an OPTIONS request may include Allow: DELETE
4.2.3	MUST	PASS	An LDPRm must not support POST

4.2.4	MUST	PASS	An LDPRm must not support PUT
4.2.5	MUST	PASS	An LDPRm must not support PATCH
4.2.6	MUST	PASS	LDPRm resources must support DELETE if DELETE is advertised in OPTIONS
4.3	MAY	SKIPPED	Although an LDPCv is both a TimeMap and an LDPC, implementations MAY disallow POST requests.
4.3.1-A	MUST	PASS	LDPCv must support GET, as is the case for any LDPR
4.3.1-B	MUST	PASS	LDPCv contain TimeMap type link header.
4.3.1-C	MUST	PASS	An LDPCv must respond to GET Accept: application/link-format as indicated in [RFC7089] section 5 and specified in [RFC6690] section 7.3.
4.3.1-D	MUST	PASS	LDPCv resources must include the Allow header
4.3.1-E	MUST	PASS	If an LDPCv supports POST, then it must include the Accept-Post header
4.3.1-F	MUST	PASS	If an LDPCv supports PATCH, then it must include the Accept-Patch header
4.3.1-G	MUST	PASS	An LDPCv, being a container must have a "Link: <http://www.w3.org/ns/ldp#Container>;rel="type""
4.3.2-A	MUST	PASS	LDPCv (version containers) MUST support OPTIONS.
4.3.2-B	MUST	PASS	LDPCv's response to an OPTIONS request MUST include "Allow: GET, HEAD, OPTIONS" per LDP
4.3.2-C	MAY	SKIPPED	LDPCv (version containers) MAY support DELETE.
4.3.2-D	MAY	SKIPPED	LDPCv (version containers) MAY support PATCH.
4.3.2-E	MAY	PASS	LDPCv (version containers) MAY support POST.
4.3.2-F	MUST	PASS	If an LDPCv supports POST, the response to an OPTIONS request MUST include the "Accept-Post" header
4.3.2-G	MUST	PASS	If an LDPCv supports PATCH, the response to an OPTIONS request MUST include the "Accept-Patch" header
4.3.3.1-A	SHOULD	PASS	If an LDPCv of an LDP-RS supports POST, a POST request that does not contain a Memento-Datetime header should be understood to create a new LDPRm contained by the LDPCv, reflecting the state of the LDPRv at the time of the POST.
4.3.3.1-B	SHOULD	PASS	If an LDPCv of an LDP-NR supports POST, a POST request

			that does not contain a Memento-Datetime header should be understood to create a new LDPRm contained by the LDPCv, reflecting the state of the LDPRv at the time of the POST.
4.3.3.1-C	MUST	PASS	If an LDPCv of an LDP-RS supports POST, a POST request that does not contain a Memento-Datetime header MUST ignore any request body.
4.3.3.1-D	MUST	PASS	If an LDPCv of an LDP-NR supports POST, a POST request that does not contain a Memento-Datetime header MUST ignore any request body.
4.3.3.1-E	SHOULD	PASS	If an LDPCv supports POST, a POST with a Memento-Datetime header should be understood to create a new LDPRm contained by the LDPCv, with the state given in the request body.
4.3.3.1-F	SHOULD	PASS	If an LDPCv supports POST, a POST with a Memento-Datetime header should be understood to create a new LDPRm contained by the LDPCv, with the datetime given in the Memento-Datetime request header.
4.3.3.2	MUST	PASS	If an implementation does not support one or both of POST cases above, it must respond to such requests with a 4xx range status code and a link to an appropriate constraints document
4.3.4	MAY	PASS	Implementations MAY disallow PUT.
4.3.5	MAY	PASS	Implementations MAY disallow PATCH
4.3.6-A	MAY	SKIPPED	An implementation MAY support DELETION of LDPCvs.
4.3.6-B	SHOULD	PASS	An implementation that does support DELETE should do so by both removing the LDPCv and removing the versioning interaction model from the original LDPRv.
5.0-A	MUST	PASS	An authorization may list any number of individual agents (that are being given access) by using the acl:agent predicate
5.0-B	MUST	PASS	An authorization may list any number of individual agents (that are being given access) by using the acl:agent predicate.
5.0-C-1	MUST	PASS	To give access to a group of agents, use the acl:agentGroup predicate. The object of an agentGroup statement is a link to a Group Listing document. The group members are listed in it, using the vcard:hasMember predicate.
5.0-C-2	MUST	PASS	To give access to a group of agents, use the acl:agentGroup predicate. The object of an agentGroup statement is a link with a hash URI to a Group Listing document. The group

			members are listed in it, using the vcard:hasMember predicate.
5.0-D	MUST	PASS	To specify that you're giving a particular mode of access to everyone, you can use acl:agentClass foaf:Agent to denote that you're giving access to the class of all agents (the general public).
5.0-E	MUST	PASS	To specify that you're giving a particular mode of access to all authenticated users, you can use acl:agentClass acl:AuthenticatedAgent to denote that you're giving access to the class of all authenticated agents.
5.0-F	MUST	PASS	The acl:accessTo predicate specifies which resources you're giving access to, using their URLs as the subjects.
5.0-G-1	MUST	PASS	acl:Read gives access to a class of operations that can be described as "Read Access". In a typical REST API, this includes access to HTTP verbs HEAD.
5.0-G-2	MUST	PASS	acl:Read gives access to a class of operations that can be described as "Read Access". In a typical REST API, this includes access to HTTP verbs GET.
5.0-H	MUST	PASS	acl:Read gives access to a class of operations that can be described as "Read Access". In a typical REST API, this includes access to HTTP verbs GET. Its absence must prevent reads
5.0-I	MUST	PASS	acl:Write gives access to a class of operations that can modify the resource. In a REST API context, this would include PUT.
5.0-J	MUST	PASS	acl:Write gives access to a class of operations that can modify the resource. In a REST API context, this would include POST.
5.0-K	MUST	PASS	acl:Write gives access to a class of operations that can modify the resource. In a REST API context, this would include DELETE
5.0-L	MUST	PASS	acl:Write gives access to a class of operations that can modify the resource. In a REST API context, this would include PATCH.
5.0-M-1	MUST	PASS	acl:Write gives access to PUT a resource. When not present, writes should be disallowed.
5.0-M-2	MUST	PASS	acl:Write gives access to POST a resource. When not present, writes should be disallowed.
5.0-M-3	MUST	PASS	acl:Write gives access to DELETE a resource. When not present, writes should be disallowed.

5.0-M-4	MUST	PASS	acl:Write gives access to PATCH a resource. When not present, writes should be disallowed.
5.0-N	MUST	PASS	acl:Append gives a more limited ability to write to a resource -- Append-Only. This generally includes the HTTP verb POST.
5.0-O	MUST	PASS	acl:Append gives a more limited ability to write to a resource -- Append-Only. This generally includes the INSERT-only portion of SPARQL-based PATCHes.
5.0-P	MUST	PASS	acl:Append gives a more limited ability to write to a resource -- Append-Only. This generally includes the HTTP verb POST, although some implementations may also extend this mode to cover non-overwriting PUTs, as well as the INSERT-only portion of SPARQL-based PATCHes. Its absence must prevent append updates.
5.0-Q	MUST	PASS	acl:Control is a special-case access mode that gives an agent the ability to view the ACL of a resource.
5.0-R	MUST	PASS	acl:Control is a special-case access mode that gives an agent the ability to modify the ACL of a resource.
5.0-S	MUST	PASS	acl:Control is a special-case access mode that gives an agent the ability to modify the ACL of a resource.
5.0-T	MUST	PASS	acl:Control is a special-case access mode that gives an agent the ability to view and modify the ACL of a resource. Its absence must prevent viewing the ACL.
5.0-U	MUST	PASS	acl:Control is a special-case access mode that gives an agent the ability to view and modify the ACL of a resource. Its absence must prevent updating the ACL.
5.0-V	MUST	PASS	acl:Control is a special-case access mode that gives an agent the ability to view and modify the ACL of a resource. Its absence must prevent updating the ACL.
5.1	MUST	PASS	An ACL for a controlled resource on a conforming server must itself be an LDP-RS.
5.2-A	MUST	PASS	Implementations must inspect the ACL RDF for authorizations. Authorizations are identified by type definition triples of the form authorization_N rdf:type acl:Authorization, where authorization_N is the URI of an authorization.
5.2-B	MUST	PASS	Implementations must use only statements associated with an authorization in the ACL RDF to determine access, except in

			the case of acl:agentGroup statements where the group listing document is dereferenced.
5.2-C	MUST	PASS	Implementations must use only statements associated with an authorization in the ACL RDF to determine access, except in the case of acl:agentGroup statements where the group listing document is dereferenced.
5.2-D	MUST	PASS	The authorizations must be examined to see whether they grant the requested access to the controlled resource.
5.2-E	MUST	PASS	If none of the authorizations grant the requested access then the request must be denied.
5.3-A	MUST	PASS	A conforming server must advertise the individual resource ACL for every controlled resource in HTTP responses with a rel="acl" link in the Link header, whether or not the ACL exists.
5.3-B	MUST	PASS	A conforming server must advertise the individual resource ACL for every controlled resource in HTTP responses with a rel="acl" link in the Link header, whether or not the ACL exists.
5.3-C	SHOULD	PASS	The ACL resource should be located in the same server as the controlled resource.
5.4-A	MAY	SKIPPED	A client HTTP POST or PUT request to create a new LDPR may include a rel="acl" link in the Link header referencing an existing LDP-RS to use as the ACL for the new LDPR.
5.4-B	MUST	PASS	The server must reject the request and respond with a 4xx or 5xx range status code, such as 409 (Conflict) if it isn't able to create the LDPR with the specified LDP-RS as the ACL. In that response, the restrictions causing the request to fail must be described in a resource indicated by a rel="http://www.w3.org/ns/ldp#constrainedBy" link in the Link response header
5.5-A	MAY	SKIPPED	Implementations may restrict support for ACLs to local resources.
5.5-B	MUST	SKIPPED	If an implementation chooses to reject requests concerning remote ACLs, it must respond with a 4xx range status code.
5.5-C	MUST	SKIPPED	If an implementation chooses to reject requests concerning remote ACLs, it must advertise the restriction with a rel="http://www.w3.org/ns/ldp#constrainedBy" link in the Link response header.
5.6-A	MAY	SKIPPED	Implementations may restrict support for groups of agents to local Group Listing documents.

5.6-B	MUST	SKIPPED	If an implementation chooses to reject requests concerning remote Group Listings, it must respond with a 4xx range status code.
5.6-C	MUST	SKIPPED	If an implementation chooses to reject requests concerning remote Group Listings, it must advertise the restriction with a rel="http://www.w3.org/ns/ldp#constrainedBy" link in the Link response header.
5.7.1-A	MUST	PASS	When a client has acl:Append but not acl:Write for an LDP-RS they MUST not DELETE, not PATCH that deletes triples, not PUT on the resource
5.7.1-B	MUST	PASS	When a client has acl:Append but not acl:Write for an LDP-RS and the implementation supports PUT to create they MUST allow the addition of a new child resource.
5.7.1-C	SHOULD	PASS	When a client has acl:Append but not acl:Write for an LDP-RS they SHOULD allow a PATCH request that only adds triples.
5.7.2	MUST	PASS	When a client has acl:Append but not acl:Write for an LDPC they MUST allow a POST request.
5.7.3	MUST	PASS	When a client has acl:Append but not acl:Write for an LDP-NR they MUST deny all DELETE, POST, and PUT requests.
5.8-A	MUST	PASS	When an ACL includes an acl:accessToClass statement, it MUST give access to all resources with the specified type, whether that type is client-managed or server-managed
5.8-B	MAY	SKIPPED	Implementations may use inference to infer types not present in a resource's triples or rel="type" links in the Link header
5.9-A	MUST	PASS	Inheritance of ACLs in Fedora implementations is defined by the [SOLIDWEBAC]ACL Inheritance Algorithm and must be reckoned along the [LDP] containment relationships linking controlled resources
5.9-B	SHOULD	PASS	In the case that the controlled resource is uncontained and has no ACL, or that there is no ACL at any point in the containment hierarchy of the controlled resource, then the server must supply a default ACL.
5.9-C	SHOULD	PASS	The default ACL resource should be located in the same server (host and port) as the controlled resource.
6.1	MUST	PASS	For every resource whose state is changed as a result of an HTTP operation, there MUST be a corresponding notification made available describing that change.

6.2-A	MUST	PASS	The notification serialization MUST conform to the [activitystreams-core] specification, and each event MUST contain the IRI of the resource and the event type.
6.2-B	SHOULD	PASS	Wherever possible, data SHOULD be expressed using the [activitystreams-vocabulary].

Release: 1.0-SNAPSHOT | #a40ce88b (2018-11-29T15:52:34Z)