OSLC Core Partial Update

New here? Sign up! [Sign up]

- Recent changes
- Core home
- All pages
- Categories
- Random Page
- Uploaded Files
- RSS
- Basic syntax guide
- Linking and categories syntax

Sort of a ghost town here
Active specification development is now at Core TC [Core TC]

Want to contribute?

1. Register [Register]
2. Complete Members Agreement [Complete Members Agreement]
3. Complete WPA [Complete WPA]

Mailing list [Mailing list]
Workgroup information [Workgroup information]

History [History] View [View] Links to this page [Links to this page]

---

Open Services for Lifecycle Collaboration
Partial Update Specification Version 2.0

Status: Working Draft

This Version

- http://open-services.net/wiki/core/OSLC-Core-Partial-Update

Latest Version

- http://open-services.net/wiki/core/OSLC-Core-Partial-Update

Previous Version

- This is the first version of this specification.

Authors

- Steve Speicher [Steve Speicher]

Contributors

- See Contributors section below

Table of Contents

Contents

- Open Services for Lifecycle Collaboration
- Partial Update Specification Version 2.0
- Introduction
- Terminology
- Use Cases & Requirements
- HTTP PATCH
  - Patch Model
    - Insert triples about a resource
      - Example - Simple insertion
    - Delete triples from a resource
      - Example - Simple delete triples
    - Updating the object of triples
      - Example - Simple update
      - Example - update blank nodes (link label)
      - Example - delete the link label
  - Patch Document
- Appendix A: Notices and References
- Contributors
- Reporting Issues on the Specification
This work is licensed under a Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/us/).

**Introduction**

There are many motivating cases for supporting a writable Web, existing approaches have focused heavily on the usage of HTTP PUT as the method for updating resources. As experience has shown, this method for updating resources comes with some significant limitations and performance costs. This has motivated the creation of the IETF Proposed Standard titled “PATCH Method for HTTP” (http://tools.ietf.org/html/rfc5789) which introduces a new HTTP verb called PATCH. RFC5789 defines the semantics of the new HTTP verb PATCH but does not endorse a patch representation or a patch model for Linked Data resources.

Motivated by our use cases and requirements, partial update can be simple. Given a resource URL, triples can be added or removed.

This document recommends to use HTTP PATCH to implement partial update, provides a patch model based on a subset of SPARQL Update. SPARQL-Update (http://www.w3.org/TR/sparql11-update/) includes a recommended PATCH document formats based on existing RDF formats and provides examples of how this approach works. If additional use cases need to be satisfied, a full SPARQL Update solution can be used.

**Terminology**

Partial Update - The HTTP operation to modify only a subset of the triples for a given resource.

Patch - See Partial update

Patch document - Representation of the patch request indicating which updates (insertions or deletions) should be performed on a given resource.

**Use Cases & Requirements**

Use Cases:

- Given a URI for a Change Request resource, I’d like to add a link to a related Test Plan. If the link already exists, then do nothing.
- Given a URI for a Test Case resource, I’d like to remove all the blocking Defect links with the set of the Defect links I have (from a picked list of existing Defects from various providers).
- Given a URI for a Test Case resource, I’d like to remove only one blocking Defect link whether it exists or not.
- Given a URI for a Change Request resource, I’d like to update the link label (blank node) associated with the related Test Plan link.

Requirements:

- Add triples to an existing resource
- Delete triples from a resource
- Updating the object of a triple in a resource
- Update reified statements on links, such as link labels
- Guidance on collision detection
- Reduce possibility of errors in updates because partial updates are safer than complete replace operations
- Reduce bandwidth used and load on server because partial updates are smaller

**HTTP PATCH**

**Patch Model**

OSLC clients **SHOULD** use the HTTP PATCH method to apply a partial update. If a server does not support PATCH on a resource, then it **SHOULD** respond with HTTP status code 405 (Method not allowed). If a server does support PATCH on a resource, then it **SHOULD** respond to a HTTP HEAD or OPTIONS operation with at least the token value "PATCH" for the response header Allow.

For servers that support partial update but not the HTTP PATCH verb, an OSLC server **MAY** accept HTTP POST requests with the HTTP header X-Method-Override: PATCH to indicate that the request is a HTTP PATCH.

OSLC Partial Update servers **MUST** treat a single HTTP PATCH request as a single resource update operation and not as independent insert and delete operations.

OSLC Partial Update servers **MUST** treat HTTP PATCH request, on an existing resource identified by the canonical Request-URI, as an atomic modification to that resource. The patch model is limited to resources that are identified by their URI that typically maps to the subject of triples. HTTP PATCH of a resource is symmetric with the usage HTTP verbs of GET, PUT and DELETE which operate on a resource identified by the Request-URI.

OSLC Partial Update servers **MAY** choose to allow the creation of new resources using HTTP PATCH. HTTP POST is the **RECOMMENDED** method for creating new resources.

OSLC clients **SHOULD** use the HTTP if-Match header and HTTP ETags to ensure it is not modifying a resource that has changed since the client last retrieved its representation. OSLC servers **SHOULD** require the HTTP if-Match header and HTTP ETags to detect collisions. OSLC servers **MUST** respond with status code 412 (Condition Failed) if ETags fail to match if there are no other errors with the request. RFC2616 (http://tools.ietf.org/html/rfc2616)

A pre-defined named graph that represents the nodes to match or to insert is sufficient.

The semantics of the PATCH model are based on SPARQL Update. Namely the subset of SPARQL Update is limited to:

- **Subset of INSERT DATA** (http://www.w3.org/TR/sparql11-update/#insertData)
- **Subset of DELETE DATA** (http://www.w3.org/TR/sparql11-update/#deleteData)

This model does not handle updating resources across multiple named graphs. Separate HTTP PATCH requests to each resource’s graph would need to be performed or utilizing a different method, such as SPARQL Update.

**Insert triples about a resource**

Named graph URI: http://open-services.net/ns/core#insertGraph

To insert new triples to a resource (graph), an OSLC client issues an HTTP PATCH request to the resource identified by the canonical Request-URI and request content of a Patch Document. The Patch Document indicates the insertion command by naming the graph http://open-services.net/ns/core#insertGraph and providing the triples to be inserted within that graph. In this case we use the SPARQL Update INSERT DATA command to specify the statements (triples) wish to insert.

OSLC Partial Update servers **MUST** support inserting triples using the within the http://open-services.net/ns/core#insertGraph named graph following the semantics of INSERT DATA command as defined by SPARQL Update (http://www.w3.org/TR/sparql11-update/).

Example - Simple insertion

Here’s an example a client request to add new property values of dcterms:description and ex:status to a file or directory resource with URL of http://example.com/tasks/27.
Before PATCH


Patch Document - Named Graph

```sparql
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix ex: <http://example.com/ns/terms#>.
@prefix os: <http://open-services.net/ns/core#>.

<http://open-services.net/ns/core#insertGraph> {
  <http://example.com/tasks/27> dcterms:description "Need to update Website owners as of May 2012" .
}
```

Patch Document - LDPatch

```sparql
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix ex: <http://example.com/ns/terms#>.

Add <http://example.com/tasks/27> dcterms:description "Need to update Website owners as of May 2012" .
```

After PATCH

```sparql
```

SPARQL Update mapping

ToDo: Provide mapping details

Example - Inserting new link label

And here's an example of adding a new dcterms:contributor value to a resource with URL of http://example.com/tasks/27. Note that the property has a value that is a blank-node of type foaf:Person. Also note that the subject of some of the triples is not the resource, but instead a blank-node reference.

```sparql
<http://open-services.net/ns/core#insertGraph> {
  _:bnode1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/person> .
  _:bnode1 <http://xmlns.com/foaf/0.1/name> "Fernando Jones" .
}
```

SPARQL Update mapping

ToDo: Provide mapping details

Delete triples from a resource

Named graph URI: http://open-services.net/ns/core#deleteGraph

To delete triples from a resource, a client will HTTP PATCH the in Patch Document (see examples below) to the resource URI with HTTP Header `Content-Type: application/x-trig'. In this case we use the SPARQL Update DELETE DATA command to specify the statements (triples) wish to remove.

OSLC server implementations that support for Partial Update MUST support removing triples using the DELETE DATA command as defined by SPARQL Update.

Example - Simple delete triples

Here's an example of what a client would patch to remove a triple from the resource at URL http://example.com/tasks/27.

Before PATCH

```sparql
```

Patch Document - Named Graph

```sparql
@prefix ex: <http://example.com/ns/terms#>.

<http://open-services.net/ns/core#deleteGraph> {
}
```

Patch Document - LDPatch

```sparql
@prefix ex: <http://example.com/ns/terms#>.

```

After PATCH

```sparql
```

SPARQL Update mapping

ToDo: Provide mapping details

Updating the object of triples

To update objects of triples for a resource, a client will send a HTTP PATCH request (see below) to the resource Request-URI with HTTP Header Content-type: application/x-trig or Content-type: application/trix+xml. In this case we use a SPARQL Update DELETE DATA command followed by an INSERT DATA command to specify the triples to be updated, the old triples and the new triples.

OSLC Partial Update servers MUST support updating triples using the DELETE DATA and INSERT DATA commands as defined by SPARQL Update.

OSLC Partial Update servers MAY allow only the the update of only those triples who’s subject is the HTTP Request-URI.

Note: this approach does NOT depend on blank node labels, which are specific RDF formats.

Example - Simple update
Before PATCH

TBD

Patch Document

@prefix ex: <http://example.com/ns/terms#>.
@prefix oslc: <http://open-services.net/ns/core#>.
@prefix oslc_cm: <http://open-services.net/ns/cm#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix dcterms: <http://purl.org/dc/terms/>.

before


After PATCH

TBD

SPARQL Update mapping

ToDo: Provide mapping details

Example - update blank nodes (link label)

Before PATCH

#prefix ex: <http://example.com/bugtracker> .
#prefix oslc: <http://open-services.net/ns/core#> .
#prefix oslc_cm: <http://open-services.net/ns/cm#> .
#prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
#prefix dcterms: <http://purl.org/dc/terms/> .

<http://example.com/bugs/2314> a oslc_cm:ChangeRequest ;
  dcterms:identifier "00002314" ;
  oslc:shortTitle "Bug 2314" ;
  dcterms:title "Invalid installation instructions" ;
  oslc:instanceShape <http://example.com/shapes/defect> ;
  dcterms:description "Invalid installation instructions indicating invalid patches to be applied," ;
  oslc:discussedBy <http://example.com/bugs/2314/discussion> ;
  oslc_cm:relatedChangeRequest <http://myserver/mycmapp/bugs/1235> ,
    <http://remoteserver/mycmapp/defects/abc123> ;
  ex:priority "Medium" ;
  ex:severity "Normal" .

_:b1 dcterms:title "A bad link title";
  rdf:object <http://myserver/mycmapp/bugs/1235>;
  rdf:predicate oslc_cm:relatedChangeRequest;
  rdf:subject <http://example.com/bugs/2314>;
  a rdf:Statement.

After PATCH

TBD

Patch Document - Named Graph

#prefix ex: <http://example.com/bugtracker> .
#prefix oslc: <http://open-services.net/ns/core#> .
#prefix oslc_cm: <http://open-services.net/ns/cm#> .
#prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
#prefix dcterms: <http://purl.org/dc/terms/> .

oslc:deleteGraph {
  _:b1 dcterms:title "A bad link title";
  rdf:object <http://myserver/mycmapp/bugs/1235>;
  rdf:predicate oslc_cm:relatedChangeRequest;
  rdf:subject <http://example.com/bugs/2314>;
  a rdf:Statement.
}

oslc:insertGraph {
  _:b1 dcterms:title "A very good link title";
  rdf:object <http://myserver/mycmapp/bugs/1235>;
  rdf:predicate oslc_cm:relatedChangeRequest;
  rdf:subject <http://example.com/bugs/2314>;
  a rdf:Statement.
}

Patch Document - LDPatch

#prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
#prefix dcterms: <http://purl.org/dc/terms/> .
#prefix oslc_cm: <http://open-services.net/ns/cm#> .


Add ?label dcterms:title "A very good link title" .

After PATCH
@prefix ex: <http://example.com/bugtracker> .
@prefix oslc: <http://open-services.net/ns/core#> .
@prefix oslc_cm: <http://open-services.net/ns/cm#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<http://example.com/bugs/2314> a oslc_cm:ChangeRequest ;
dcterms:identifier "00002314" ;
oslc:shortTitle "Bug 2314" ;
dcterms:title "Invalid installation instructions" ;
oslc:instanceShape <http://example.com/shapes/defect> ;
dcterms:description "Invalid installation instructions indicating invalid patches to be applied." ;
oslc:discussedBy <http://example.com/bugs/2314/discussion> ;
oslc_cm:relatedChangeRequest <http://myserver/mycmapp/bugs/1235> ,
<http://remoteserver/mycmapp/defects/abc123> ;
ex:priority "Medium" ;
ex:severity "Normal" .

_mapped to SPARQL Update

prefix ex: <http://example.com/bugtracker> .
prefix oslc: <http://open-services.net/ns/core#> .
prefix oslc_cm: <http://open-services.net/ns/cm#> .

DELETE {?s dcterms:title ?o .}
INSERT {?s dcterms:title "A very good link title" .}
WHERE {
  ?s rdf:predicate oslc_cm:relatedChangeRequest .
}

Example - delete the link label

Before PATCH

PREFIX oslc: <http://open-services.net/ns/core#> .
PREFIX oslc_cm: <http://open-services.net/ns/cm#> .
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://example.com/bugs/2314> a oslc_cm:ChangeRequest .

_mapped to SPARQL Update

PREFIX oslc: <http://open-services.net/ns/core#> .
PREFIX oslc_cm: <http://open-services.net/ns/cm#> .
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

oslc:deleteGraph {?
  _x14 dcterms:title "A very good link title";
  rdf:object <http://myserver/mycmapp/bugs/1235>;
  rdf:predicate oslc_cm:relatedChangeRequest;
  rdf:subject <http://example.com/bugs/2314>;
  a rdf:Statement.
}

Patch Document - Named Graph

PREFIX oslc: <http://open-services.net/ns/core#> .
PREFIX oslc_cm: <http://open-services.net/ns/cm#> .
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

oslc:deleteGraph {
  _x14 dcterms:title "A very good link title";
  rdf:object <http://myserver/mycmapp/bugs/1235>;
  rdf:predicate oslc_cm:relatedChangeRequest;
  rdf:subject <http://example.com/bugs/2314>;
  a rdf:Statement.
}

Patch Document - LDPatch

PREFIX oslc_cm: <http://open-services.net/ns/cm#> .
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .


Delete dcterms:title "A very good link title" .

After PATCH

PREFIX oslc: <http://open-services.net/ns/core#> .
PREFIX oslc_cm: <http://open-services.net/ns/cm#> .
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://example.com/bugs/2314> a oslc_cm:ChangeRequest .

Mapped to SPARQL Update

Note: a clever server may decide to remove the additional reification triples if decides there are no other interesting facts that they hold.

**Patch Document**

To use HTTP PATCH a client must provide with the request a Patch Document format that is able convey the change to be match in the partial update. This guidance recommends the use of named graph (quad) formats for the patch document. In this way, it does not require a new format or parsers to be created. Some quad formats available include TriG (Turtle-based) and TriX (XML-based).

**Proposed RESOLUTION:** Since Core 2.0 requires RDF/XML it would make sense that those implementations would use TriX. For Core 3.0 (W3C LDP based), which requires Turtle, it would make sense that those implementations would use TriG (plus n-triples).

**Appendix A: Notices and References**

**Contributors**
- Steve Speicher (IBM, OSLC-Core Lead)
- Arthur Ryman (IBM, OSLC-Core)
- Nick Crossley (IBM, OSLC-Core)

**Reporting Issues on the Specification**

The working group participants who author and maintain this working draft specification, monitor a distribution list where issues or questions can be raised, see Core Mailing List. Also the issues found with this specification and their resolution can be found at Core Issues.

**Change History**

- 2014-09-24 Include examples that leverage LD Patch Format from current working draft (SteveS)

**License and Intellectual Property**

We make this specification available under the terms and conditions set forth in the site Terms of Use, IP Policy, and the Workgroup Participation Agreement for this Workgroup.

**References**

- OSLC Core - OSLC Core Specification 2.0
- Dublin Core 1.1 - Dublin Core Metadata Element Set, Version 1.1
- FOAF - Friend of a Friend (FOAF)
- HTTP PATCH - PATCH Method for HTTP
- JSON - JavaScript Object Notation
- RDF/XML Syntax - RDF / XML Syntax Specification (Revised)
- URI Syntax - URI Generic Syntax
- XML Namespaces - Namespaces in XML 1.0 (Third Edition)
- TurtlePatch
- Linked Data Basic Profile - W3C Member Submission
- SPARQL 1.1 Graph Store HTTP Protocol - PATCH
- Talis changeset proposal

**Category:** Supporting Documents

**Categories**

- Supporting Documents

All content Creative Commons Attribution 3.0 US unless otherwise specified. See more terms of use.