OASIS OPEN PROJECTS

OSLC Tracked Resource Set Version 3.0. Part 1: Specification

OASIS Standard
23 July 2023

This stage:

https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/os/tracked-resource-set.html (Authoritative)
https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/os/tracked-resource-set.pdf

Previous stage:

https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/ps02/tracked-resource-set.html (Authoritative)
https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/ps02/tracked-resource-set.pdf

Latest stage:

https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/tracked-resource-set.html (Authoritative)
https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/tracked-resource-set.pdf

Latest version:

https://open-services.net/spec/trs/latest

Latest editor's draft:

https://open-services.net/spec/trs/latest-draft

Open Project:

OASIS Open Services for Lifecycle Integration (OSLC) Open Project

Project Chairs:

Jim Amsden (jamsden@us.ibm.com), IBM
Andrii Berezovskyi (andriib@kth.se), KTH

Editor:

Nick Crossley (nick_crossley@us.ibm.com), IBM

Additional components:

This specification is one component of a Work Product that also includes:

- *OSLC TRS Version 3.0. Part 1: Specification (this document)*. https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/os/tracked-resource-set.html
- *OSLC TRS Version 3.0. Part 2: Vocabulary*. https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/os/tracked-resource-set-vocab.html
- *OSLC TRS Version 3.0. Part 3: Constraints*. https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/os/tracked-resource-set-shapes.html
- *OSLC TRS Version 3.0: Part 4: Machine-readable RDF Vocabulary*. https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/os/trs-vocab.ttl
- *OSLC TRS Version 3.0: Part 5: Machine-readable Resource Shapes*. https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/os/trs-shapes.ttl

RDF Namespaces:

http://open-services.net/ns/core/trs#

http://open-services.net/ns/core/trspatch#

 Copyright © OASIS Open 2023. All Rights Reserved.

Abstract:

The Tracked Resource Set protocol allows a server to expose a set of resources in a way that allows clients to discover that set of resources, to track additions to and removals from the set, and to track state changes to the resources in the set. The protocol does not assume that clients will dereference the resources, but they could do so. The protocol is suitable for dealing with sets containing a large number of resources, as well as highly active resource sets that undergo continual change. The protocol is HTTP-based and follows RESTful principles.

Status:

This document was last revised or approved by the membership of OASIS on the above date. The level of approval is also listed above. Check the "Latest stage" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Open Project are listed at https://open-services.net/about/.

Comments on this work can be provided by opening issues in the project repository or by sending email to the project's public comment list oslc-op@lists.oasis-open-projects.org.

The English version of this specification is the only normative version. Non-normative translations may also be available. Note that any machine-readable content (Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Citation format:

When referencing this specification the following citation format should be used:

[OSLC-TRS-v3.0]

OSLC Tracked Resource Set Version 3.0. Part 1: Specification. Edited by Nick Crossley. 23 July 2023. OASIS Standard. https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/os/tracked-resource-set.html. Latest stage: https://docs.oasis-open-projects.org/oslc-op/trs/v3.0/tracked-resource-set.html.

 Copyright © OASIS Open 2023. All Rights Reserved.

Notices

Copyright © OASIS Open 2023. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This specification is published under the Attribution 4.0 International (CC BY 4.0). Portions of this specification are also provided under the Apache License 2.0.

All contributions made to this project have been made under the OASIS Contributor License Agreement (CLA).

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Open Projects IPR Statements page.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Open Project or OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Project Specification or OASIS Standard, to notify the OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Open Project that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Open Project Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark/ for above guidance.

Table of Contents

 Copyright © OASIS Open 2023. All Rights Reserved.

1. Introduction

This section is non-normative.

An OSLC Tracked Resource Set (TRS) provides a mechanism for making a set of resources discoverable and for reporting ongoing changes affecting the set. This allows tools to expose a live feed of linked lifecycle data in a way that permits other tools to monitor the tracked resources, aggregate information from multiple sets of tracked resources, and maintain live, searchable information based on that linked data.

1.1 Terminology

Terminology is based on OSLC Core Overview [OSLCCore3], W3C Linked Data Platform [LDP], W3C's Architecture of the World Wide Web [WEBARCH], and Hyper-text Transfer Protocol [HTTP11].

Tracked Resource Set (TRS)

Describes a resource that defines a finite, enumerable collection of Tracked Resources expressed as a Base and a Change Log.

Tracked Resource

A resource, identified by URI, that is a member of one or more Tracked Resource Sets.

Base

The portion of a Tracked Resource Set representation that lists the Tracked Resources at some point in time.

Change Log

The portion of a Tracked Resource Set representation detailing a series of Change Events for Tracked Resources, where those changes are relative to the Base.

Change Event

Identifies an addition, removal, or state change of a Tracked Resource in a Tracked Resource Set. These events are represented using the three RDF classes `trs:Creation`, `trs:Modification`, and `trs:Deletion` - but note that the distinction between creation and modification is historical and servers are not required to observe any such distinction.

TRS Patch

An extended Change Event in a Tracked Resource Set detailing a change to the resource's RDF representation.

TRS Server

An application or application component that provides one or more Tracked Resource Sets.

1.2 Typographical Conventions and Use of RFC Terms

As well as sections marked as non-normative, all authoring guidelines, diagrams, examples, and notes in this specification are non-normative. Everything else in this specification is normative.

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this specification are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Sample resource representations are provided in `text/turtle` format [Turtle].

The following common URI prefixes are used throughout this specification:

```
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix ex:      <http://example.org/>.
@prefix ldp:     <http://www.w3.org/ns/ldp#>.
@prefix oslc:    <http://open-services.net/ns/core#>.
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#>.
```

 Copyright © OASIS Open 2023. All Rights Reserved.

2. Basic Concepts

This section is non-normative.

A TRS Server maintains one or more Tracked Resource Sets. The TRS Server decides which particular resources are in a particular Tracked Resource Set at any moment; both the Tracked Resource Sets and the linked data contents of each Tracked Resource can vary over time. A Tracked Resource Set is represented in terms of a Base and a Change Log. The Base is a Linked Data Platform [LDP] Container providing a point-in-time enumeration of the Tracked Resource members of the Tracked Resource Set. The Change Log provides a (possibly empty) ordered series of adjustments describing changes to the Tracked Resources. When the Base is empty, the Change Log describes a history of how the Tracked Resource Set has grown and evolved since its inception. A TRS Server can periodically update the Base of a TRS and truncate the Change Log to avoid excessively large Change Logs. Clients can read the Base and poll the Change Log to derive and maintain an up-to-date picture of the state of all the Tracked Resources in a Tracked Resource Set.

The Change Log can contain earlier Change Event entries that would be accounted for by the Base portion. A “cutoff” property of the Base identifies the point in the Change Log at which processing of Change Events can be cut off because this change and older changes are already included in the Base. TRS Clients use the Base to establish an initial set of resources to track, and the Change Log to address changes to that set, and changes to the those resources.

 Copyright © OASIS Open 2023. All Rights Reserved.

3. Discovery

> NOTE: Discovery is a convenience
>
> Discoverability is a convenience; an administrator can configure a Client with a particular Tracked Resource Set knowing just the URI of the Server's Tracked Resource Set. Documentation for an TRS Server can describe its Tracked Resource Sets, including the URI of each of the Server's Tracked Resource Set resources.

A Server **MAY** provide multiple Tracked Resource Sets, and **MAY** make its Tracked Resource Sets discoverable. [trs-1]

A `trs:trackedResourceSet` property **MAY** be used to declare the whereabouts of a Tracked Resource Set resource; the value of such a property **MUST** be the URI of a Tracked Resource Set. [trs-2]

This allows the existence and location of a Tracked Resource Set resource to be declared with an RDF statement like the following:

> EXAMPLE 1: Discovery Statement

```
@prefix trs: <http://open-services.net/ns/core/trs#> .
<> trs:trackedResourceSet <https://a.example.com/trs1> .
```

Where such RDF statements might be found is outside the scope of this specification. See also [OSLCCore3], section "Well-known URI Bootstrapping".

 Copyright © OASIS Open 2023. All Rights Reserved.

4. General Behavior

The TRS Server **MUST** support GET requests conformant with [OSLCCore2] or [OSLCCore3] for Tracked Resource Sets, Base resources, and Change Logs. [trs-3]

The RDF returned for Tracked Resource Sets, Base resources, Change Logs, **MUST** conform to the shapes and constraints in the following section. [trs-4]

The TRS Server **SHOULD** support ETags, caching, and conditional GETs for those resources. [trs-5]

Servers providing the tracked resources referenced by a Tracked Resource Set **MUST** also support GET requests conformant with [OSLCCore2] or [OSLCCore3] and **SHOULD** support ETags, caching, and conditional GETs for those resources. [trs-6]

 Copyright © OASIS Open 2023. All Rights Reserved.

5. Vocabulary Terms and Constraints

OSLC TRS Version 3.0. Part 2: Vocabulary and OSLC TRS Version 3.0. Part 3: Constraints define the vocabulary terms and constraints for OSLC Change Management resources. These terms and constraints are specified according to [OSLCCore3].

Copyright © OASIS Open 2023. All Rights Reserved.

6. Cutoff Event

The cutoff event in the Base marks the point in the Change Log at which this change and all earlier changes have already been accounted for in the Base.

If the cutoff property is missing, or if it has the value `rdf:nil`, the Base enumerates the (possibly empty) Resource Set at the inception of the Tracked Resource Set, and the Change Log **MUST** list all changes since that inception. [trs-7]

Otherwise, the identified Change Event **MUST** be in the Change Log - that is, there **MUST NOT** be a discontinuity between the Base portion and the Change Log portion of a Tracked Resource Set. [trs-8]

Conversely, the Change Log **MAY** contain earlier Change Event entries that have been accounted for in the Base. [trs-9]

 Copyright © OASIS Open 2023. All Rights Reserved.

7. Change Logs and Creation or Modification Events

A Server **MUST** refer to a given Tracked Resource using the exact same URI in the Base membership property and every Change Event (`trs:changed` reference) for that resource. [trs-10]

A Change Log represents a series of changes to the corresponding Tracked Resource Set over some period of time. For any given tracked resource, the Change Log **MUST** contain a sufficient sequence of Change Events such that processing the Base and the Change Log gives a consistent and complete picture of the Tracked Resources at the end of that period of time. [trs-11]

The ordering of Change Log events is meaningful for changes to any specific resource, but there is no meaning to the ordering of change log events for different resources. That is, a change event for resource `R1` at time `t` **MAY** have an order number that is greater than a change to an unrelated resource `R2` at time `t+1` [trs-12].

The Change Log does not necessarily identify *every* change of state before or during that period of time, as further explained in the following paragraphs.

A Server **MAY** batch up changes and add a batch of consolidated Change Events to the Change Log at some interval. [trs-13]

A Server **MUST** eventually report a change event for a tracked resource if at time t_0, a tracked resource is added to the Tracked Resource Set, deleted from the Tracked Resource Set, or the state of the Tracked Resource Set is modified, such that a GET on that resource would now return a semantically different response from a GET request issued just before t_0. [trs-14]

When a resource is modified two or more times in rapid succession, a Server **MAY** elide such modifications by reporting only a single creation or modification event in the Change Log. [trs-15]

A Server **MAY** report a modification event even in cases where there would be no significant difference in response. [trs-16] This could happen because a resource was modified and then that change was reverted in a second modification; the first modification could be elided in the change log.

There is no difference between change events with RDF type `trs:Creation` and `trs:Modification`; the two types exist for historical reasons. When a new resource is modified or added to a tracked resource set, a Server **MAY** issue a modification event or a creation event [trs-17]; as described earlier, the server **MUST** eventually issue at least one such event [trs-18].

When a resource is created and deleted in rapid succession, a Server **MAY** omit all Change Events for that resource. [trs-19]

Because of the highly dynamic nature of the Resource Set and the difficulty of enumerating an exact set of resources at a precise point in time, a Server **MAY** produce a Base that is only an approximation of the Tracked Resource Set membership [trs-20]. A Base might omit mention of a Resource that ought to have been included or include a Resource that ought to have been omitted. For each such inconsistency in the Base, the Server **MUST** at some point include a corrective Change Event in the Change Log more recent than the base cutoff event [trs-21], allowing a client to compute the correct set of Tracked Resources.

A Server **MAY** issue a deletion event for a resource that is not a member of the Tracked Resource Set. This includes issuing multiple deletion events for (what was) a member of the Tracked Resource Set. This might happen when the base is being recalculated, and the change log slightly overlaps the base, or as part of the corrective set of change events described in the previous paragraph. [trs-22]

> NOTE
>
> From the above sections, one can infer that the consistency and completeness of the change log with respect to the data on a TRS Server is only possible (but is not guaranteed) after reading the entire Base and Change Log. A client cannot skip the initial reading of the base, nor partially consume a Change Log, and then expect consistency with some earlier point-in-time of the server's data. Even having read the entire base plus change log, a client may still not have information on the most recent changes to the tracked resources; nor is there any guarantee that the result represents an exact state of the server's data at any specific point in time.

 Copyright © OASIS Open 2023. All Rights Reserved.

8. Change Log Segmentation

The number of Change Events in the Change Log can grow to a point where it is not reasonable to contain all the Change Events in a single HTTP response. In this case, the most recent Change Event resources **MUST** be included in the RDF representation of the Tracked Resource Set itself [trs-23], and earlier events **MAY** be segmented into separate Change Log resources referenced from the `trs:previous` property [trs-24].

There can be any number of such Change Log segments. The events in a Change Log segment **MUST** all have order numbers higher than any events in any subsequent Change Log segments. [trs-25]

Just as the most recent Change Events **MUST** be included inline in the HTTP response for the Change Log in the Tracked Resource Set resource itself, TRS Servers **MUST** include the Change Event resources for each Change Log segment inline in the RDF representation of the HTTP response for the Change Log segment. [trs-26]

> NOTE
>
> The above paragraphs allows a Client to discover the most recent Change Events, retrieve successively older Change Log resources until it encounters a Change Event that has already been processed (on a previous check), and for it to be guaranteed that a segment cannot contain Change Events with a higher order number (representing a more recent change) than the events already encountered on earlier segments in the chain. The protocol does not attach significance to where a Server breaks the Change Log into separate segments, i.e., the number of entries in a `trs:ChangeLog` is entirely up to the Server.

 Copyright © OASIS Open 2023. All Rights Reserved.

9. Base Segmentation

The number of Tracked Resources in the Base could be such that it is not reasonable to list them all in a single HTTP response. In this case, the Base **MAY** be segmented into separate resources; there can be any number of such Base segments, each one listing a subset of all Tracked Resources. [trs-27]

If base paging is implemented, the server **MAY** respond with a 30x redirect message, directing the Client to the first “single-page resource”, or alternatively it **MAY** respond with the first "single-page resource" . [trs-28]

The representation of a single-page resource **MAY** contain a subset of the Base’s membership triples. [trs-29]

The response for a single-page resource **SHOULD** contain a Link: <http://www.w3.org/ns/ldp#Page>; rel="type" header. [trs-30]If there are further pages of base page members for the single-page response, the response **MUST** include a Link: <uriOfNextPage>; rel="next" header, where uriOfNextPage is the URI of the next page. [trs-31]

The first single-page resource of a Base **MUST** include a trs:cutoffEvent. [trs-32]

A Server **MAY** include a Tracked Resource in more than one base segment. [trs-33]

 Copyright © OASIS Open 2023. All Rights Reserved.

10. Updating Change Log segments, Truncating Change Logs, and Replacing the Base

A Server **MAY** add new change events at the start of the initial set of events returned inline with the Tracked Resource itself [trs-34] (in fact, this is the **only** way in which new change events are added to the Change Log).

A Server **MUST NOT** move a change event from one segment of a segmented change log to an earlier segment in the chain, one with more recent change events. [trs-35]

A Server **MAY** move a change event from one segment of a segmented change log to an later segment in the chain, one with older change events. [trs-36]

> NOTE
>
> The above implies that a client could see the same change event more than once. Since each change event has a unique URI, clients can detect such repetition, and ignore duplicate events.

A Server **MAY** remove a change event from the change log [trs-37], perhaps because that change is now reflected in an updated base with a new cutoff event.

A chain of Change Log segments **MAY** continue all the way back to the inception of the Tracked Resource Set and contain Change Events for every change made since then. [trs-38] However, to avoid maintaining this ever growing list of Change Logs indefinitely, a Server **MAY** truncate the Change Log at a suitable point in the chain [trs-39]. This can be accomplished by deleting the oldest segments of the Change Log and/or by removing `trs:previous` links that reference them, and/or by removing obsolete Change Events from the end of the Change Log.

> NOTE
>
> TRS Servers can therefore return HTTP status code 404 (Not found) when navigating the "previous" reference from a final or stale Change Log segment; clients should interpret this as indicating that they have reached the end of the change log.

To ensure that a new Client can always get started, the Change Log **MUST** contain the base cutoff event of the corresponding Base [trs-40], and all Change Events more recent than it. Thus the Server is only allowed to truncate Change Events older than the base cutoff event. When the Base has no base cutoff event (i.e., the Base enumerates the Resource Set at the inception of the Tracked Resource Set), the Change Log **MUST** contain all Change Events back to the inception of the Tracked Resource Set; i.e., no truncation is allowed [trs-41].

To allow truncation of long Change Logs, a TRS Server **MAY** calculate a new base set at any time, with a new cutoff event in the Change Log [trs-42]. When doing this, In order to handle clients that are currently processing the older Base and Change Log [trs-43]:

- When calculating a new Base, the TRS Server **SHOULD** retain Change Events before the new cutoff event that a reasonable client might still be processing. [trs-44]
- When producing multiple segments of a new Base, the TRS Server **MUST NOT** reuse URIs from segments of a previous Base. [trs-45]

 Copyright © OASIS Open 2023. All Rights Reserved.

11. Timescales and event frequency

This section is non-normative.

In order to provide adequate response to client requests, a TRS server needs to allow those clients sufficient time to read the base, the change log, and process the set of tracked resources. However, the data volumes and timescales involved in TRS processing are likely to vary between servers for different applications. A server representing Amazon transactions might have many events per second, while a server representing exhibits at a museum might have a few events per month. The cost of processing a single event is also likely to vary between applications; reading a new or modified resource with 5 RDF properties will take less time than reading one with 5,000 properties.

For these reasons, the TRS specification does not impose specific constraints over the length of time for which a TRS base needs to remain readable, nor what the degree of overlap should be between a base and a corresponding change log. A server implementing TRS needs to consider, and should document, the quality of service it will provide in terms of the size of pages in the base or change log, how long base pages are kept, how long change events are kept, and the minimum period for which change events behind the latest base cutoff are kept.

Copyright © OASIS Open 2023. All Rights Reserved.

12. TRS Examples

This section is non-normative.

In this example of a change log, time stamps are used to generate unique URNs for the Change Event URIs; other ways of generating a unique URI are also possible.

Note that the actual time of change is not included in a Change Event. Only a sequence number, representing the "sequence in time" of each change is provided.

EXAMPLE 2: Change Log

```
# Resource: http://cm1.example.com/trackedResourceSet
@prefix trs: <http://open-services.net/ns/core/trs#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://cm1.example.com/trackedResourceSet>
  a trs:TrackedResourceSet ;
  trs:base <http://cm1.example.com/baseResources/> ;
  trs:changeLog [
    a trs:ChangeLog ;
    trs:change <urn:example:6e8bc430:cm1.example.com:2010-10-27T17:39:33.000Z:103> ;
    trs:change <urn:example:6e8bc430:cm1.example.com:2010-10-27T17:39:32.000Z:102> ;
    trs:change <urn:example:6e8bc430:cm1.example.com:2010-10-27T17:39:31.000Z:101> .
  ] .

<urn:example:6e8bc430:cm1.example.com:2010-10-27T17:39:33.000Z:103>
  a trs:Creation ;
  trs:changed <http://cm1.example.com/bugs/23> ;
  trs:order "103"^^xsd:integer .

<urn:example:6e8bc430:cm1.example.com:2010-10-27T17:39:32.000Z:102>
  a trs:Modification ;
  trs:changed <http://cm1.example.com/bugs/22> ;
  trs:order "102"^^xsd:integer .

<urn:example:6e8bc430:cm1.example.com:2010-10-27T17:39:31.000Z:101>
  a trs:Deletion ;
  trs:changed <http://cm1.example.com/bugs/21> ;
  trs:order "101"^^xsd:integer .
```

 Copyright © OASIS Open 2023. All Rights Reserved.

13. TRS Patch

This section describes an mechanism allowing Change Events to carry detailed information about modifications to resources with an inline RDF representation, avoiding the need for a client to issue a GET request for a new or modified resource. This allows more efficient handling of frequent small changes to a large resources and creation of new versions of existing resources. The patch format is a subset of [RDFPatch].

A patch event **MUST** have type `trs:Modification` or `trs:Creation` Change Events; it is not meaningful for `trs:Deletion` Change Events. [trs-46]

13.1 TRS Patch

A TRS patch **MUST** consist of a sequence of directives delimited by a '.' [trs-47]

Each directive of a patch **MUST** consist of four terms. [trs-48]

Each term in a directive **MAY** be separated with white space, including newlines. [trs-49]

The first term of a directive **MUST** be the ASCII character 'A' (U+0041) or the ASCII character 'D' (U+0044)". [trs-50]

A directive with the first term being the ASCII character 'A' (U+0041) describes the addition of one RDF triple to the resource's RDF data graph. The subject, predicate, and object of the triple to be added are given in terms 2 to 4 of the directive.

A directive with the first term being the ASCII character 'D' (U+0044) describes the deletion of one RDF triple from the resource's RDF data graph. The subject, predicate, and object of the triple to be added are given in terms 2 to 4 of the directive.

The subject and predicate (terms 2 and 3) in a directive **MUST** be in the form of absolute URI references enclosed between '<' and '>' [trs-51]

The object in a directive (term 4) **MUST** be either an absolute URI reference enclosed between '<' and '>', or a literal in [Turtle] syntax. [trs-52]

The order of patch directives is significant - changes **MUST** be applied in the order given. [trs-53]

The value for property trspatch:rdfPatch must conform to the following EBNF:

```
rdfPatch       ::= (addProperty | deleteProperty)*
addProperty    ::= 'A' triple
deleteProperty ::= 'D' triple
```

Where triple is defined in RDF 1.1 N-Triples except BLANK_NODE_LABEL is not allowed for subject or object. The trspatch:rdfPatch properties are properties of either a trs:Creation or trs:Modification resource.

[trs-54]

The following is an example of a TRS patch that deletes one RDF triple (subject `http://example.com/bob`, predicate `http://xmlns.com/foaf/0.1/knows`, object `https://example.com/alice`) and adds another RDF triple (subject `https://example.com/fred`, predicate `http://http://xmlns.com/foaf/0.1/member`, object `http://example.com/old-timers`):

EXAMPLE 3: Patch to delete a triple

```
D <http://example.com/bob> <http://xmlns.com/foaf/0.1/knows> <http://example.com/alice> .
A <http://example.com/fred> <http://http://xmlns.com/foaf/0.1/member> <http://example.com/old-timers> .
```

13.2 TRS Patch Example

This section is non-normative.

Turtle representation for the resource `https://a.example.com/sw-movie/versions/1`. Assume that when the resource is retrieved in this state, the entity tag `783xhaty95` is returned in the ETag response header:

EXAMPLE 4: sw-movie Version 1

```
# The following is the representation of
# https://a.example.com/sw-movie/versions/1
# in the state with entity tag 783xhaty95
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix ldp: <http://www.w3.org/ns/ldp#>.
<https://a.example.com/sw-movie/versions/1>
   dcterms:isVersionOf <https://a.example.com/sw-movie> .
```

 Copyright © OASIS Open 2023. All Rights Reserved.

```
<https://a.example.com/sw-movie>
   a ldp:Resource;
   dcterms:title "Star Wars".
```

Turtle representation for the resource **https://a.example.com/sw-movie/versions/2**. Assume that when the resource is retrieved in this state, the entity tag **212gyysxx8** is returned in the ETag response header:

EXAMPLE 5: sw-movie Version 2

```
# The following is the representation of
# https://a.example.com/sw-movie/versions/2
# in the state with entity tag 212gyysxx8
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix ldp: <http://www.w3.org/ns/ldp#>.
<https://a.example.com/sw-movie/versions/2>
   dcterms:isVersionOf <https://a.example.com/sw-movie> .
<https://a.example.com/sw-movie>
   a ldp:Resource;
   dcterms:title "Star Wars: Episode IV - A New Hope".
```

Turtle representation for a Change Event describing the creation of the resource **https://a.example.com/sw-movie/versions/2**. The TRS patch describes the state of this new resource in terms of the state of resource **https://a.example.com/sw-movie/versions/1**:

EXAMPLE 6: Create sw-movie version 2 from version 1

```
# The following is the representation of a change event
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix trs: <http://open-services.net/ns/core/trs#>.
@prefix trspatch: <http://open-services.net/ns/core/trspatch#>.
<urn:example:6e8bc430:a.example.com:2014-11-20T13:08:00.000Z:102>
   a trs:Creation;
   trs:changed <https://a.example.com/sw-movie/version/2>;
   trs:order "192"^^xsd:integer;
   trspatch:createdFrom <https://a.example.com/sw-movie/version/1>;
   trspatch:beforeEtag "783xhaty95";
   trspatch:afterEtag "212gyysxx8";
   trspatch:rdfPatch
     """
      D <https://a.example.com/sw-movie/versions/1>  <http://purl.org/dc/terms/isVersionOf> <https://a.example.com/sw-movie> .
      A <https://a.example.com/sw-movie/versions/2>  <http://purl.org/dc/terms/isVersionOf> <https://a.example.com/sw-movie> .
      D <https://a.example.com/sw-movie> <http://purl.org/dc/terms/title> "Star Wars".
      A <https://a.example.com/sw-movie> <http://purl.org/dc/terms/title> "Star Wars: Episode IV - A New Hope".
     """.
```

 Copyright © OASIS Open 2023. All Rights Reserved.

14. Conformance

This specification is based on [OSLCCore3]. OSLC Tracked Resource Set Servers servers **MUST** be compliant with the Core specification, **MUST** follow all the mandatory requirements in the normative sections of each part of this specification, and **SHOULD** follow all the guidelines and recommendations in both these specifications. [trs-55]

Clause Number	Requirement
trs-1	A Server **MAY** provide multiple Tracked Resource Sets, and **MAY** make its Tracked Resource Sets discoverable.
trs-2	A `trs:trackedResourceSet` property **MAY** be used to declare the whereabouts of a Tracked Resource Set resource; the value of such a property **MUST** be the URI of a Tracked Resource Set.
trs-3	The TRS Server **MUST** support GET requests conformant with [OSLCCore2] or [OSLCCore3] for Tracked Resource Sets, Base resources, and Change Logs.
trs-4	The RDF returned for Tracked Resource Sets, Base resources, Change Logs, **MUST** conform to the shapes and constraints in the following section.
trs-5	The TRS Server **SHOULD** support ETags, caching, and conditional GETs for those resources.
trs-6	Servers providing the tracked resources referenced by a Tracked Resource Set **MUST** also support GET requests conformant with [OSLCCore2] or [OSLCCore3] and **SHOULD** support ETags, caching, and conditional GETs for those resources.
trs-7	If the cutoff property is missing, or if it has the value `rdf:nil`, the Base enumerates the (possibly empty) Resource Set at the inception of the Tracked Resource Set, and the Change Log **MUST** list all changes since that inception.
trs-8	Otherwise, the identified Change Event **MUST** be in the Change Log - that is, there **MUST NOT** be a discontinuity between the Base portion and the Change Log portion of a Tracked Resource Set.
trs-9	Conversely, the Change Log **MAY** contain earlier Change Event entries that have been accounted for in the Base.
trs-10	A Server **MUST** refer to a given Tracked Resource using the exact same URI in the Base membership property and every Change Event (`trs:changed` reference) for that resource.
trs-11	For any given tracked resource, the Change Log **MUST** contain a sufficient sequence of Change Events such that processing the Base and the Change Log gives a consistent and complete picture of the Tracked Resources at the end of that period of time.
trs-12	That is, a change event for resource `R1` at time `t` **MAY** have an order number that is greater than a change to an unrelated resource `R2` at time `t+1`
trs-13	A Server **MAY** batch up changes and add a batch of consolidated Change Events to the Change Log at some interval.
trs-14	A Server **MUST** eventually report a change event for a tracked resource if at time t_0, a tracked resource is added to the Tracked Resource Set, deleted from the Tracked Resource Set, or the state of the Tracked Resource Set is modified, such that a GET on that resource would now return a semantically different response from a GET request issued just before t_0.
trs-15	When a resource is modified two or more times in rapid succession, a Server **MAY** elide such modifications by reporting only a single creation or modification event in the Change Log.
trs-16	A Server **MAY** report a modification event even in cases where there would be no significant difference in response.
trs-17	When a new resource is modified or added to a tracked resource set, a Server **MAY** issue a modification event or a creation event
trs-18	the server **MUST** eventually issue at least one such event
trs-19	When a resource is created and deleted in rapid succession, a Server **MAY** omit all Change Events for that resource.
trs-20	a Server **MAY** produce a Base that is only an approximation of the Tracked Resource Set membership
trs-21	For each such inconsistency in the Base, the Server **MUST** at some point include a corrective Change Event in the Change Log more recent than the base cutoff event
trs-22	A Server **MAY** issue a deletion event for a resource that is not a member of the Tracked Resource Set. This includes issuing multiple deletion events for (what was) a member of the Tracked Resource Set. This might happen when the base is being recalculated, and the change log slightly overlaps the base, or as part of the corrective set of change events described in the previous paragraph.
trs-23	the most recent Change Event resources **MUST** be included in the RDF representation of the Tracked Resource Set itself
trs-24	earlier events **MAY** be segmented into separate Change Log resources referenced from the `trs:previous` property
trs-25	There can be any number of such Change Log segments. The events in a Change Log segment **MUST** all have order numbers higher than any events in any subsequent Change Log segments.
trs-26	Just as the most recent Change Events **MUST** be included inline in the HTTP response for the Change Log in the Tracked Resource Set resource itself, TRS Servers **MUST** include the Change Event resources for each Change Log segment inline in the RDF representation of the HTTP response for the Change Log segment.

 Copyright © OASIS Open 2023. All Rights Reserved.

Clause Number	Requirement
trs-27	the Base **MAY** be segmented into separate resources; there can be any number of such Base segments, each one listing a subset of all Tracked Resources.
trs-28	If base paging is implemented, the server **MAY** respond with a 30x redirect message, directing the Client to the first “single-page resource”, or alternatively it **MAY** respond with the first "single-page resource" .
trs-29	The representation of a single-page resource **MAY** contain a subset of the Base’s membership triples.
trs-30	The response for a single-page resource **SHOULD** contain a Link: <http://www.w3.org/ns/ldp#Page>; rel="type" header.
trs-31	If there are further pages of base page members for the single-page response, the response **MUST** include a Link: <uriOfNextPage>; rel="next" header, where uriOfNextPage is the URI of the next page.
trs-32	The first single-page resource of a Base **MUST** include a trs:cutoffEvent.
trs-33	A Server **MAY** include a Tracked Resource in more than one base segment.
trs-34	A Server **MAY** add new change events at the start of the initial set of events returned inline with the Tracked Resource itself
trs-35	A Server **MUST NOT** move a change event from one segment of a segmented change log to an earlier segment in the chain, one with more recent change events.
trs-36	A Server **MAY** move a change event from one segment of a segmented change log to an later segment in the chain, one with older change events.
trs-37	A Server **MAY** remove a change event from the change log
trs-38	A chain of Change Log segments **MAY** continue all the way back to the inception of the Tracked Resource Set and contain Change Events for every change made since then.
trs-39	a Server **MAY** truncate the Change Log at a suitable point in the chain
trs-40	the Change Log **MUST** contain the base cutoff event of the corresponding Base
trs-41	When the Base has no base cutoff event (i.e., the Base enumerates the Resource Set at the inception of the Tracked Resource Set), the Change Log **MUST** contain all Change Events back to the inception of the Tracked Resource Set; i.e., no truncation is allowed
trs-42	a TRS Server **MAY** calculate a new base set at any time, with a new cutoff event in the Change Log
trs-43	In order to handle clients that are currently processing the older Base and Change Log
trs-44	When calculating a new Base, the TRS Server **SHOULD** retain Change Events before the new cutoff event that a reasonable client might still be processing.
trs-45	When producing multiple segments of a new Base, the TRS Server **MUST NOT** reuse URIs from segments of a previous Base.
trs-46	A patch event **MUST** have type `trs:Modification` or `trs:Creation` Change Events; it is not meaningful for `trs:Deletion` Change Events.
trs-47	A TRS patch **MUST** consist of a sequence of directives delimited by a ‘.’
trs-48	Each directive of a patch **MUST** consist of four terms.
trs-49	Each term in a directive **MAY** be separated with white space, including newlines.
trs-50	The first term of a directive **MUST** be the ASCII character 'A' (U+0041) or the ASCII character 'D' (U+0044)".
trs-51	The subject and predicate (terms 2 and 3) in a directive **MUST** be in the form of absolute URI references enclosed between ‘<’ and ‘>’
trs-52	The object in a directive (term 4) **MUST** be either an absolute URI reference enclosed between ‘<’ and ‘>’, or a literal in [Turtle] syntax.
trs-53	The order of patch directives is significant - changes **MUST** be applied in the order given.

trs-54	The value for property trspatch:rdfPatch must conform to the following EBNF:
	
	`rdfPatch ::= (addProperty | deleteProperty)*`
	`addProperty ::= 'A' triple`
	`deleteProperty ::= 'D' triple`
	
	Where triple is defined in RDF 1.1 N-Triples except BLANK_NODE_LABEL is not allowed for subject or object. The trspatch:rdfPatch properties are properties of either a trs:Creation or trs:Modification resource.

trs-55	This specification is based on [OSLCCore3]. OSLC Tracked Resource Set Servers servers **MUST** be compliant with the Core specification, **MUST** follow all the mandatory requirements in the normative sections of each part of this specification, and **SHOULD** follow all the guidelines and recommendations in both these specifications.

 Copyright © OASIS Open 2023. All Rights Reserved.

Appendix A. Acknowledgements

This section is non-normative.

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

James Amsden, IBM (Chair, Editor)
Frank Budinsky, IBM
Nick Crossley, IBM
Vivek Garg, IBM
Ian Green, IBM
Arthur Ryman, IBM
Steve Speicher, IBM

 Copyright © OASIS Open 2023. All Rights Reserved.

Appendix B. References

B.1 Normative references

[LDP]

Steve Speicher; John Arwe; Ashok Malhotra. *Linked Data Platform 1.0*. W3C, 26 February 2015. W3C Recommendation. URL: https://www.w3.org/TR/ldp/

[OSLCCore3]

Steve Speicher; Jim Amsden. *OSLC Core 3.0*. OASIS. URL: https://docs.oasis-open-projects.org/oslc-op/core/v3.0/oslc-core.html

[RFC2119]

S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF, March 1997. Best Current Practice. URL: https://www.rfc-editor.org/rfc/rfc2119

[RFC8174]

B. Leiba. *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*. IETF, May 2017. Best Current Practice. URL: https://www.rfc-editor.org/rfc/rfc8174

[Turtle]

Eric Prud'hommeaux; Gavin Carothers. *RDF 1.1 Turtle*. W3C, 25 February 2014. W3C Recommendation. URL: https://www.w3.org/TR/turtle/

B.2 Informative references

[HTTP11]

R. Fielding, Ed.; J. Reschke, Ed.. *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*. IETF, June 2014. Proposed Standard. URL: https://httpwg.org/specs/rfc7230.html

[OSLCCore2]

S. Speicher; D. Johnson. *OSLC Core 2.0*. http://open-services.net. Finalized. URL: http://open-services.net/bin/view/Main/OslcCoreSpecification

[RDFPatch]

Andy Seaborne; Rob Vesse. *RDF Patch – Describing Changes to an RDF Dataset*. Unofficial Draft 20 November 2014. URL: http://afs.github.io/rdf-patch/

[WEBARCH]

Ian Jacobs; Norman Walsh. *Architecture of the World Wide Web, Volume One*. W3C, 15 December 2004. W3C Recommendation. URL: https://www.w3.org/TR/webarch/

 Copyright © OASIS Open 2023. All Rights Reserved.