OASISOPENPROJECTS

OSLC Core Version 3.0. Part 3: Resource Preview

Project Specification 01
17 September 2020

This stage:

https://docs.oasis-open-projects.org/oslc-op/core/v3.0/ps01/resource-preview.html (Authoritative)
https://docs.oasis-open-projects.org/oslc-op/core/v3.0/ps01/resource-preview.pdf

Previous stage:

https://docs.oasis-open-projects.org/oslc-op/core/v3.0/psd04/resource-preview.html (Authoritative)
https://docs.oasis-open-projects.org/oslc-op/core/v3.0/psd04/resource-preview.pdf
(published as Project Specification Draft on 20 December 2019)

Latest stage:

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

Latest version:

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

Latest editor's draft:

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

Open Project:

OASIS Open Services for Lifecycle Collaboration (OSLC) OP

Project Chairs:

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

Editor:

Jim Amsden (jamsden@us.ibm.com), IBM

Additional components:

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

- *OSLC Core Version 3.0. Part 1: Overview.* oslc-core.html
- *OSLC Core Version 3.0. Part 2: Discovery.* discovery.html
- *OSLC Core Version 3.0. Part 3: Resource Preview* (this document). resource-preview.html
- *OSLC Core Version 3.0. Part 4: Delegated Dialogs.* dialogs.html
- *OSLC Core Version 3.0. Part 5: Attachments.* attachments.html
- *OSLC Core Version 3.0. Part 6: Resource Shape.* resource-shape.html
- *OSLC Core Version 3.0. Part 7: Vocabulary.* core-vocab.html
- *OSLC Core Version 3.0. Part 8: Constraints.* core-shapes.html
- *OSLC Core Version 3.0. Part 9: Machine Readable Vocabulary Terms.* core-vocab.ttl
- *OSLC Core Version 3.0. Part 10: Machine Readable Constraints.* core-shapes.ttl

Related work:

This specification is related to:

- *OSLC Core Version 3.0: Link Guidance*. https://oslc-op.github.io/oslc-specs/notes/link-guidance.html

RDF Namespaces:

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

Abstract:

Resource Preview defines a technique to get a minimal HTML representation of a resource identified by a URL. Applications often use this capability to render or display links with more appropriate icons and labels, or display a preview when a user mouses over a link.

Status:

This document was last revised or approved by the OASIS Open Services for Lifecycle Collaboration (OSLC) OP 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.

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-ResourcePreview-3.0]
OSLC Core Version 3.0. Part 3: Resource Preview. Edited by Jim Amsden. 17 September 2020. OASIS Project Specification 01. https://docs.oasis-open-projects.org/oslc-op/core/v3.0/ps01/resource-preview.html. Latest stage: https://docs.oasis-open-projects.org/oslc-op/core/v3.0/resource-preview.html.

Notices

Table of Contents

1. Introduction

This section is non-normative.

This specification describes how a client application can display links and embed rich previews for resources from other applications. Links may have a label and an icon, and previews are HTML markup provided by a server and displayed directly inside the client application.
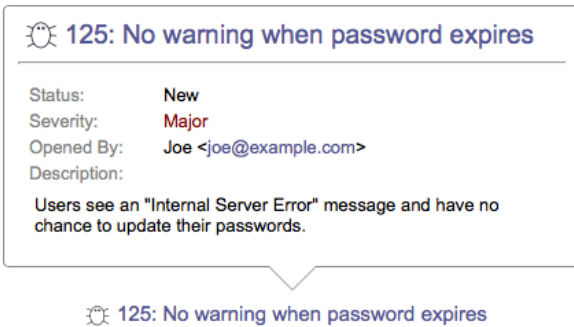
Fig. 1 Preview for a Link

Previews often appear as a pop-up when a user mouses over a link as in Fig. 1 Preview for a Link. However, a client may wish to display a preview differently depending on the kind of application, the size of the screen, and the capabilities of the device. For example, a desktop application on a PC might handle previews differently than a mobile application running on a small touchscreen.

Servers can provide both small and large previews. A client might show a small preview first, then if the user gestures, show additional details from the large preview. Servers suggest sizes for previews, and previews can ask to be resized after they are displayed.

A client only needs to know the URI of a resource to display a link and a preview. It doesn't need to know anything else about the resource. Clients don't need to copy, synchronize, or cache any data.

1.1 Terminology

Terminology uses and extends the terminology and capabilities of OSLC Core Overview, W3C Linked Data Platform [LDP], W3C's Architecture of the World Wide Web [WEBARCH], and Hyper-text Transfer Protocol [HTTP11].

The following terms are used in discussions of previews:

Compact resource

A resource describing how to display a link and preview for another, associated resource.

Preview

An HTML representation of a resource that can be embedded in another user interface.

1.2 References

1.2.1 Normative references

[CSS21]

Bert Bos; Tantek Çelik; Ian Hickson; Håkon Wium Lie et al. *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*. 7 June 2011. W3C Recommendation. URL: https://www.w3.org/TR/CSS2/

[HTTP11]

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

[JSON-LD]

Manu Sporny; Gregg Kellogg; Markus Lanthaler. *JSON-LD 1.0*. 16 January 2014. W3C Recommendation. URL: https://www.w3.org/TR/json-ld/

[JSONSchema]

F. Galiegue; K. Zyp. *JSON Schema*. Informational. URL: http://tools.ietf.org/html/draft-zyp-json-schema-04

[LDP]

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

[OSLCUIPreview20]

S. Bosworth. *OSLC Core 2.0 - UI Preview*. Finalized. URL: http://open-services.net/bin/view/Main/OslcCoreUiPreview

[RFC2119]

S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. March 1997. Best Current Practice. URL: https://tools.ietf.org/html/rfc2119

[RFC4627]

D. Crockford. *The application/json Media Type for JavaScript Object Notation (JSON)*. July 2006. Informational. URL: https://tools.ietf.org/html/rfc4627

[RFC5988]

M. Nottingham. *Web Linking*. October 2010. Proposed Standard. URL: https://tools.ietf.org/html/rfc5988

[RFC7240]

J. Snell. *Prefer Header for HTTP*. June 2014. Proposed Standard. URL: https://tools.ietf.org/html/rfc7240

[turtle]

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

[webmessaging]

Ian Hickson. *HTML5 Web Messaging*. 19 May 2015. W3C Recommendation. URL: https://www.w3.org/TR/webmessaging/

1.2.2 Informative references

[WEBARCH]

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

1.3 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, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this specification are to be interpreted as described in [RFC2119].

The namespace for OSLC Core is `http://open-services.net/ns/core#`.

Commonly used namespace prefixes:

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

2. Motivation

This section is non-normative.

Applications often display links to related resources from other applications. For example, a quality management application might show a link to a related defect when displaying a test result. In some cases, the quality management application only has a URI pointing to the defect. The application needs a way to display the link with an appropriate label and icon so testers find out the status of the defect without leaving the quality management application.

Compact resources and resource preview provide these capabilities. Clients can discover the link label and a resource icon given only the resource URI. The client can then use the link label and icon as a means of displaying the link in a meaningful way. Previews allow users to see see information about related resources quickly without leaving the application they're in, even when the resources are from another server.

3. Working with Previews

This section is non-normative.

To enable previews of a resource, servers supply an associated Compact resource describing the preview. The Compact resource can contain a link label, icon, and small and/or large previews of the resource. Compact resources always have a JSON representation [RFC4627], but they can also have other representations such as XML, Turtle [turtle] or JSON-LD [JSON-LD]. Here is a simple example of a Compact resource:

Example 1: A Compact Resource

```
{
    "title": "Screenshot of the problem",
    "icon": "http://example.com/icons/attachment.jpg",
    "smallPreview": {
        "document": "http://example.com/bugs/324/screenshot?preview=small"
    }
}
```

To display a link with a label or a preview of a resource, clients request its Compact resource. The URI of the Compact resource is found through an HTTP `Link` header [RFC5988] in HTTP responses to the resource URI. Alternatively, a client can use a `Prefer` request header in requests for the resource to ask for its Compact resource. The server can then inline the Compact resource directly in the response, saving an HTTP request.

The Compact resource may contain small and/or large previews. Each preview can have size hints and a link to an HTML representation designed to be embedded in other user interfaces. To display the preview, the client creates an HTML `iframe` element in its user interface and sets the `iframe` element's `src` to the preview document URI. This sandboxes the preview from other content in the client application and allows the preview to use its own stylesheets and scripts.

3.1 Getting the Compact Resource

This section is non-normative.

Clients can discover Compact resources in three ways:

1. Discovering Compact Resources Using the HTTP Accept Header
2. Discovering Compact Resources Using the HTTP Link Header
3. Discovering Compact Resources Using the HTTP Prefer Header

The HTTP `Accept` header allows clients to access previews using the `application/x-oslc-compact+xml` MIME type extension to request the Compact resource representation for a URI instead of the content of the resource itself. This usage is considered deprecated and is included only to maintain compatibility with [OSLCUIPreview20].

The HTTP `Link` header allows servers to offer previews for any resource, even binary resources with no RDF or JSON representations. To discover previews using the `Link` header, a client typically performs an HTTP HEAD or OPTIONS request on the resource URI. The response contains a `Link` header with the URI of the Compact resource. The client then performs a GET request on the Compact URI to retrieve the Compact resource.

Clients might instead use the HTTP `Prefer` header to get the Compact resource in a single HTTP request. If the client is confident that the resource has a preview, it makes an HTTP GET request on the resource URI using the `return=representation` preference [RFC7240] and an `include` parameter [LDP] asking for the Compact resource to be included in the response. The provider responds with a minimal representation of the resource and the Compact resource in the HTTP response body.

3.1.1 Discovering Compact Resources Using the HTTP Accept Header

This section is non-normative.

[OSLCUIPreview20] utilizes a MIME type extension, `application/x-oslc-compact+xml` to allow servers to use content negotiation with the HTTP `Accept` header to access the Compact representation of a resource instead of the content of the resource itself.

Example 2: Requesting the Compact Resource

```
GET /bugs/324/screenshot HTTP/1.1
Host: example.com
Accept: application/x-oslc-compact+xml
```

This returns a Compact resource with `Content-Type: application/x-oslc-compact+xml` as defined in Appendix B. XML Representation Format.

3.1.2 Discovering Compact Resources Using the HTTP Link Header

This section is non-normative.

In responses to HTTP requests for resources that have a preview, servers include a `Link` header [RFC5988] where the link relation is `http://open-services.net/ns/core#Compact` and the target URI is the URI of the Compact resource.

Example 3: Requesting the Resource Headers

```
HEAD /bugs/324/screenshot HTTP/1.1
Host: example.com
```

Example 4: Response with a Compact Link

```
HTTP/1.1 200 OK
Content-Length: 45789
Content-Type: image/png
ETag: "678609cdee68e0fb8aea5f252b84a511"
Link: <http://example.com/bugs/324/screenshot?compact>; rel="http://open-services.net/ns/core#Compact"
```

Clients can request the Compact resource using the target URI of the `Link` header.

Example 5: Requesting the Compact Resource

```
GET /bugs/324/screenshot?compact HTTP/1.1
Host: example.com
Accept: application/json
```

Example 6: JSON Compact Response

```
HTTP/1.1 200 OK
Content-Length: 192
Content-Type: application/json; charset=UTF-8
ETag: "3bf6fbc90e11b3c2cc30acb534b452ea"
Vary: Accept,Accept-Language

{
    "title": "Screenshot of the problem",
    "shortTitle": "screenshot.png",
    "smallPreview": {
        "document": "http://example.com/bugs/324/screenshot?preview=small"
    }
}
```

Servers can also return a `Link` header in response to successful requests that create resources. This allows the client to get the Compact resource URI for new resources without making additional requests. Note that servers set the `Link` context using an `anchor` parameter if the request URI is not the same as the newly-created resource URI.

Example 7: Creating a Resource

```
POST /bugs HTTP/1.1
Host: example.com
Content-Length: 153
Content-Type: text/turtle

@prefix dcterms: <http://purl.org/dc/terms/> .

<>      a               <http://example.com/ns#Bug> ;
        dcterms:title  "Something went wrong" .
```

Example 8: Creation Response with a Compact Link

```
HTTP/1.1 201 Created
Content-Length: 0
Link: <http://example.com/bugs/478?compact>; rel="http://open-services.net/ns/core#Compact"; anchor="http://example.com/bugs/478"
Location: http://example.com/bugs/478
```

3.1.3 Discovering Compact Resources Using the HTTP Prefer Header

This section is non-normative.

Clients can request a Compact resource by making an HTTP GET request to the target resource's URI using the `return=representation` preference of the HTTP Prefer request header [RFC7240] and `include` parameter [LDP] value `http://open-services.net/ns/core#PreferCompact`. Servers supporting resource preview must support this method of discovery for resources with RDF or JSON representations.

Example 9: Requesting the Compact Resource Using the Prefer Header

```
GET /bugs/324 HTTP/1.1
Host: example.com
Accept: application/json
```

```
Prefer: return=representation; include="http://open-services.net/ns/core#PreferCompact"
```

If the provider supports a preview for this resource and the request is successful, the response includes the Compact resource in its body.

The response content type is negotiated using the HTTP `Accept` request header. If the negotiated content type is `application/json`, the response body is a JSON object. The top-level JSON object has a `compact` property whose value is the JSON object describing the Compact resource in Appendix A. JSON Representation Format. The JSON may include other properties.

Example 10: Preference-Applied Response

```
HTTP/1.1 200 OK
Content-Length: 315
Content-Type: application/json; charset=UTF-8
ETag: "f9d76afe5fbed1655c5768906db8958a"
Preference-Applied: return=representation
Vary: Accept,Accept-Language,Prefer

{
    "compact": {
        "title": "324: Need a fix <em>NOW</em>",
        "icon": "http://example.com/icons/defect.jpg",
        "largePreview": {
            "document": "http://example.com/bugs/324?preview=large",
            "hintHeight": "250px",
            "hintWidth": "400px"
        }
    }
}
```

Services may include a JSON-LD context in an `application/json` response. Clients who prefer RDF should request `text/turtle` or `application/ld+json` using the HTTP `Accept` request header, rather than `application/json`.

Example 11: JSON Response with a JSON-LD Context

```
HTTP/1.1 200 OK
Content-Length: 788
Content-Type: application/json; charset=UTF-8
ETag: "d53d19be87fa9c61043c70bd91413dab"
Preference-Applied: return=representation
Vary: Accept,Accept-Language,Prefer

{
    "@id": "http://example.com/bugs/324",
    "@type": "http://example.com/ns#Bug",
    "@context": "http://docs.oasis-open.org/oslc-core/oslc-core/v3.0/cs01/contexts/preview.jsonld",
    "compact": {
        "@id": "http://example.com/bugs/324?compact",
        "@type": "http://open-services.net/ns/core#Compact",
        "title": "324: Need a fix <em>NOW</em>",
        "shortTitle": "324",
        "icon": "http://example.com/icons/defect.jpg",
        "iconTitle": "Defect",
        "iconAltLabel": "Defect",
        "largePreview": {
            "document": "http://example.com/bugs/324?preview=large",
            "hintHeight": "250px",
            "hintWidth": "400px"
        },
        "smallPreview": {
            "document": "http://example.com/bugs/324?preview=small"
        }
    }
}
```

3.2 Displaying Previews

This section is non-normative.

When displaying a preview inside another HTML-based presentation, clients create an `iframe` element, setting its `src` attribute to the preview's document URI. Previews can contain HTML, stylesheets, and scripts and might not render properly outside of an `iframe`. Using the `iframe` element also sandboxes the preview, avoiding cross-site scripting vulnerabilities.
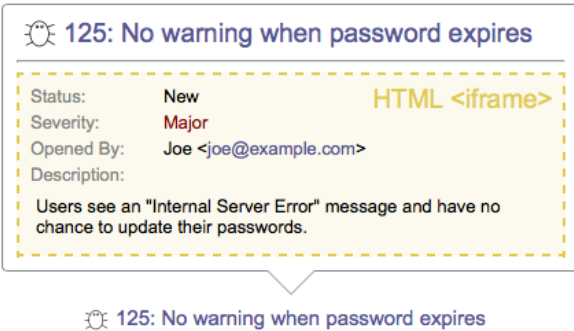
Fig. 2 Preview iframe

Typically, the server does not include the icon or title in the preview itself. The client can display these as needed. In Fig. 2 Preview iframe, the preview document is the content inside the yellow, dashed lines.

Resources can have both a small and large preview, and clients can choose which to display. Some clients display the small preview initially and let users expand to the large preview using a button or "Show more" link.

The Compact resource can include `oslc:hintHeight` and `oslc:hintWidth` properties for each preview. Values for `oslc:hintWidth` and `oslc:hintHeight` are expressed in *length units* specified in [CSS21].

Servers can also request dynamic resizing for small and large previews. The communication of dynamic size information happens within a web browser. JavaScript code running in the preview `iframe` sends a message that is received and acted upon by JavaScript code running in the `iframe`'s parenting window. This cross-frame communication is done using HTML 5 `postMessage`. Dynamic resizing will not work inside browsers that do not support `postMessage`.

A resize message begins with `oslc-resize:` followed by a JSON object describing the dimensions using properties `oslc:hintHeight` and `oslc:hintWidth`.

JSON Property Name	Type	Occurs	Description
`oslc:hintHeight`	String	Zero-or-one	Preferred height of the preview. Values are expressed using length units as specified in [CSS21].
`oslc:hintWidth`	String	Zero-or-one	Preferred width of the preview. Values are expressed using length units as specified in [CSS21].

For example, the following message requests that the preview be resized to height of 277 pixels and a width of 400 pixels.

Example 12: Dyanmic Resize Message

```
oslc-resize:{"oslc:hintHeight": "277px", "oslc:hintWidth": "400px"}
```

The following JavaScript example sends a resize request to the parent window.

Example 13: JavaScript postMessage Example

```
var size = {
    'oslc:hintHeight': '277px',
    'oslc:hintWidth': '400px'
};

if (window.postMessage && window.parent)  {
    window.parent.postMessage('oslc-resize:' + JSON.stringify(size), "*");
}
```

Clients can ignore a server's size hints and use other values. For instance, a client might choose another size if the preview is too large for the window.

4. Preview Guidance

This section is non-normative.

The following guidance is suggested for Client display of a resource link and rich previews. For purposes of this discussion, assume source resource A has a URI property that refers to target resource B:

4.1 Navigation to the target resource

This section is non-normative.

At any time, the user should be able to gesture that they would like to navigate the link to the target resource, regardless of how the link to the target resource happens to be rendered.

4.2 Default way to display the link to the target resource

This section is non-normative.

The typical way for forming and displaying the hyperlink should be based on information that is stored locally in source resource A (or that is generally known to the Client). Generally, link text can be derived from property values of resource A, and potentially from a string or literal property value in the reference to the target resource B, if such a property exists. Because these property values are part of the representation of the source resource A, they are available without consulting the target resource, and will be available even if the target resource B cannot be fetched. When available, Clients should display this string (as opposed to the URI) when presenting resource A to indicate a potential navigation to resource B. This is the basic presentation of a link to target resource B.

The default display of the link from A to B is visible to any user authorized to access resource A. The use of Compact representation information described below is only viable for users who are also authorized to access resource B.

4.3 Using the Compact representation title, short title, and icon

This section is non-normative.

A Compact representation of the resource may contain a more accurate and slightly richer label for a target resource (`dcterms:title` element), a short-form title for the resource (`oslc:shortTitle`), and a corresponding image (`oslc:icon`), possibly chosen from a set with different sizes (`oslc:iconSrcSet`), all of which may be based on the current state of the target resource. If this becomes known to the Client, the client should assume that this information is better and use it in forming the hyperlink that is displayed to the user. When available, `oslc:shortTitle` may be used instead of `dcterms:title` in presentations where visual space is severely limited.

Clients should not fetch Compact representations when there is perfectly usable default display information available. When designing a Client application, consideration should be given to the potential benefits of obtaining an improved title and icon for the target resource against the costs of preemptively fetching the Compact representation in terms of added load on servers and networks (which might only be apparent to users on slow networks or heavily loaded servers).

Clients should be wary of material obtained from non-trusted sources; in particular, the Client should check that the dcterms:title and oslc:shortTitle property values do not contain HTML markup beyond the specified set of simple elements.

4.4 Preview sizing

This section is non-normative.

When the resource does not supply preferred sizing for a preview, the Client should default to a reasonably generous value. The default value a Client uses may be different for different window and screen sizes.

4.5 Displaying a small preview

This section is non-normative.

If a user mouses or hovers over a displayed link, the Client should determine whether the target resource has a small preview (`oslc:smallPreview`). If it does, the Client should present the small preview document in a hovering widget. Since preview documents may contain arbitrary content, including HTML and scripts, Client must use iframes if embedding the preview document inside another HTML-based presentation.

The Client should not attempt to prefetch a Compact representation just to have the preview URIs in hand so that the hover can come up faster. There is a low chance that the user will make a gesture that would call for the display of a small preview. It would generally be a poor trade-off to increase overall system load just to decrease UI latency for low probability eventualities. Clients may wish to utilize lazy loading techniques to only access preview dialogs that users actually view, and can cache those dialogs for subsequent uses if needed to improve performance.

4.6 Displaying a large preview

This section is non-normative.

If the user then gestures that they want to see a bit more of the resource, the Client should determine whether the target resource has a large preview (`oslc:largePreview`). If it does, the Client should present the large preview document in a stationary widget that permits further user interaction. Again, since preview documents may contain arbitrary content, including HTML and scripts, Clients must use iframes if embedding the preview document inside another HTML-based presentation.

5. Implementation Conformance

5.1 General

Servers **MAY** choose to provide Compact resources for some resources and not others. [rp-1]

Resources with previews **MUST** support the HTTP OPTIONS method. [rp-2]

In responses to HTTP GET requests targeting resources that have a Compact resource, servers **SHOULD** either include a `Vary` response header with at least `Accept` and `Prefer` field values or a `Cache-Control` header value `no-store`. [rp-3]

Servers **MAY** consult the `Accept-Language` header on requests for a Compact resource (and on requests for preview documents) to return a resource for the requested natural language. [rp-4]

Clients **SHOULD** gracefully handle unsuccessful attempts to use previews. Previews may not be supported for all resources or may not work due to security or service availability issues. [rp-5]

5.2 Accept Header

In response to OPTIONS, HEAD or GET on resource, servers **MAY** include an `Accept: application/x-oslc-compact+xml` header to indicate the resource provides a compact representation. [rp-6]

Servers **MAY** support the `application/x-oslc-compact+xml` MIME type extension to allow clients to use content negotiation to access the Compact resource. However, this usage should be considered deprecated and is only provided for OSLC 2.0 compatibility. [rp-7]

Servers that do support the `application/x-oslc-compact+xml` MIME type extension **MUST** return the Compact resource using `Content-Type: application/x-oslc-compact+xml` as specified in [OSLCUIPreview20]. [rp-8]

5.3 Link Headers

In responses to successful HTTP requests for a target resource that has a Compact resource, servers **MUST** include a `Link` header [RFC5988] where:

- The context URI is the effective request URI,
- The link relation is `http://open-services.net/ns/core#Compact`, and
- The target URI is the URI of the Compact resource.

[rp-9]

Example 14: Example Link Header

```
Link: <http://example.com/bugs/324/screenshot?compact>; rel="http://open-services.net/ns/core#Compact"
```

If a newly-created resource has a Compact resource, servers **MAY** return a `Link` header in response to the creation request where:

- The context URI is the URI of the newly-created resource,
- The link relation is `http://open-services.net/ns/core#Compact`, and
- The target URI is the URI of the associated Compact resource.

[rp-10]

Example 15: Example Link Header with Anchor

```
Link: <http://example.com/bugs/478?compact>; rel="http://open-services.net/ns/core#Compact"; anchor="http://example.com/bugs/478"
```

5.4 Prefer Headers

Clients **MAY** request that the Compact resource is returned inline with the target resource using the `Prefer` request header [RFC7240] and

- Preference `return`, value `representation`
- Parameter `include` [LDP], value `http://open-services.net/ns/core#PreferCompact`.

[rp-11]

Example 16: Example Prefer Header

```
Prefer: return=representation; include="http://open-services.net/ns/core#PreferCompact"
```

Servers **MUST** honor a client's request to in-line the Compact resource in JSON and RDF representations if the target resource has a Compact

resource and the request is successful. [rp-12]

If the target resource does not exist or is not accessible to the requesting client, servers **MUST** return the same status code that it would have returned had the client not included the `Prefer` header in the request. [rp-13]

When servers in-line the Compact resource with the target resource in an `application/json` [RFC4627] response, the response entity **MUST** be a JSON object with a `compact` property where the value is the Compact resource JSON as described in Appendix A. JSON Representation Format. [rp-14]

Servers **MAY** choose to return only a subset of the target resource properties when a client requests that the Compact resource is returned in-line. [rp-15]

Clients **SHOULD** inspect the response body for the Compact resource even if the server omits the `Preference-Applied` header [RFC7240]. [rp-16]

5.5 Compact Resources

Servers **MUST** support the `application/json` [RFC4627] and `text/turtle` [turtle] media types for the Compact resource. [rp-17]

Servers **SHOULD** support the `application/ld+json` media type [JSON-LD] for the Compact resource. [rp-18]

The `application/json` Compact resource representation **MUST** be in the JSON format described in Appendix A. JSON Representation Format. [rp-19]

Servers **MAY** respond with JSON-LD in compacted document form when clients request the `application/json` representation of the Compact resource as long as the response meets the requirements for the JSON representation. [rp-20]

RDF representations of the Compact resource **MUST** conform to the shape specified in 6.1 Resource: Compact. [rp-21]

Content-Type `application/x-oslc-compact+xml` representations of the Compact resource **MUST** conform to the shape specified in Appendix B. XML Representation Format. [rp-22]

The Compact resource **MAY** have additional server-specific properties. [rp-23]

5.6 Previews

When displaying a preview inside another HTML presentation, clients **MUST** use an `iframe` element, setting its `src` attribute to the preview's document URI. [rp-24]

Servers **MUST** express the `oslc:hintWidth` and `oslc:hintHeight` properties of an `oslc:Preview` in length units as specified in [CSS21]. [rp-25]

Servers **MAY** use dynamic resizing for small and large previews in addition to `oslc:Preview oslc:hintHeight` and `oslc:hintWidth` properties. [rp-26]

Servers that support dynamic sizing **MUST** send resize messages to the parent window using the `Window.postMessage` method [webmessaging] where the source of the event is the preview's window. [rp-27]

A resize message **MUST** consist of the characters `oslc-resize:` followed by a JSON object with at least one of the following properties:

- `oslc:hintHeight` or
- `oslc:hintWidth`.

[rp-28]

Example 17: Example Resize Message

```
oslc-resize:{"oslc:hintHeight": "277px", "oslc:hintWidth": "400px"}
```

A resize message **MAY** also also consist of the characters `oslc-preview-height:` followed by the desired height in pixels. [rp-29]

Example 18: Example Resize Height Message

```
oslc-preview-height:277
```

This `postMessage` format should be considered deprecated and is only provided for OSLC 2.0 compatibility.

The `oslc:hintHeight` and `oslc:hintWidth` property values in the resize message JSON **MUST** be length units as specified in [CSS21]. [rp-30]

Servers **MAY** send more than one dynamic resize request. [rp-31]

Clients **MAY** choose to ignore the size hints in `oslc:Preview` resources or in dynamic resize messages. [rp-32]

6. Resource Constraints

This document applies the following constraints to the [OSLC Core vocabulary](#) terms.

<mark>An OSLC server providing the Resource Preview capability **MUST** implement the vocabulary defined in this section. [rp-33]</mark>

6.1 Resource: Compact

- **Describes:** `http://open-services.net/ns/core#Compact`
- **Summary:** Describes how to display a resource preview.

Compact Properties

Prefixed Name	Occurs	Read-only	Value-type	Representation	Range	Description
`dcterms:title`	Zero-or-one	true	string	N/A	Unspecified	Title that may be used in the display of a link to the resource. The value should include only content that is valid inside an HTML <span> element. Providers should include a dcterms:title property with an informative label for the resource. The title is typically shown to a user as a hyperlink. For a resource with no obvious title, Providers should omit the dcterms:title property. Providers must first HTML escape the contents of the dcterms:title before sending the response.
`oslc:icon`	Zero-or-one	true	Resource	Reference	Unspecified	URI of an image which may be used in the display of a link to the resource.
`oslc:iconAltLabel`	Zero-or-one	true	string	N/A	Unspecified	Alternative label used in association with the oslc:icon, such as HTML img tag's alt attribute.
`oslc:iconSrcSet`	Zero-or-one	true	string	N/A	Unspecified	Specification of a set of images of different sizes based on HTML img element srcset attribute.
`oslc:iconTitle`	Zero-or-one	true	string	N/A	Unspecified	Title used in association with the oslc:icon, such as HTML img tag's title attribute.
`oslc:largePreview`	Zero-or-one	true	AnyResource	Either	`oslc:Preview`	URI and sizing properties for an HTML document to be used for a large preview.
`oslc:shortTitle`	Zero-or-one	true	string	N/A	Unspecified	Abbreviated title which may be used in the display of a link to the resource. The value should include only content that is valid inside an HTML <span> element. Providers should include an abbreviated title for the resource when possible. The abbreviated title is typically shown to a user as a hyperlink in presentations where visual space is limited. As a general guideline, the length of the abbreviated title should be 5 characters or less. A user-visible identifier that ordinarily appears in the dcterms:title, such as a defect number, makes for a good oslc:shortTitle value. When a resource has no obvious identifier or handle, Providers should omit the oslc:shortTitle property. Providers must first HTML escape the contents of the oslc:shortTitle before sending the response.
`oslc:smallPreview`	Zero-or-one	true	AnyResource	Either	`oslc:Preview`	URI and sizing properties for an HTML document to be used for a small preview.

6.2 Resource: Preview

- **Describes:** `http://open-services.net/ns/core#Preview`
- **Summary:** An HTML representation of a resource that can be embedded in another user interface.

Preview Properties

Prefixed Name	Occurs	Read-only	Value-type	Representation	Range	Description
`oslc:document`	Exactly-one	true	Resource	Reference	Unspecified	The URI of an HTML document to be used for the preview.
`oslc:hintHeight`	Zero-or-one	true	string	N/A	Unspecified	Recommended height of the preview. Values are expressed using length units as specified in [CSS21].
`oslc:hintWidth`	Zero-or-one	true	string	N/A	Unspecified	Recommended width of the preview. Values are expressed using length units as specified in [CSS21].

7. Conformance

Implementations of this specification need to satisfy the following conformance clauses.

Clause Number	Requirement
rp-1	Servers **MAY** choose to provide Compact resources for some resources and not others.
rp-2	Resources with previews **MUST** support the HTTP OPTIONS method.
rp-3	In responses to HTTP GET requests targeting resources that have a Compact resource, servers **SHOULD** either include a `Vary` response header with at least `Accept` and `Prefer` field values or a `Cache-Control` header value `no-store`.
rp-4	Servers **MAY** consult the `Accept-Language` header on requests for a Compact resource (and on requests for preview documents) to return a resource for the requested natural language.
rp-5	Clients **SHOULD** gracefully handle unsuccessful attempts to use previews. Previews may not be supported for all resources or may not work due to security or service availability issues.
rp-6	In response to OPTIONS, HEAD or GET on resource, servers **MAY** include an `Accept: application/x-oslc-compact+xml` header to indicate the resource provides a compact representation.
rp-7	Servers **MAY** support the `application/x-oslc-compact+xml` MIME type extension to allow clients to use content negotiation to access the Compact resource. However, this usage should be considered deprecated and is only provided for OSLC 2.0 compatibility.
rp-8	Servers that do support the `application/x-oslc-compact+xml` MIME type extension **MUST** return the Compact resource using `Content-Type: application/x-oslc-compact+xml` as specified in [OSLCUIPreview20].
rp-9	In responses to successful HTTP requests for a target resource that has a Compact resource, servers **MUST** include a `Link` header [RFC5988] where: • The context URI is the effective request URI, • The link relation is `http://open-services.net/ns/core#Compact`, and • The target URI is the URI of the Compact resource.
rp-10	If a newly-created resource has a Compact resource, servers **MAY** return a `Link` header in response to the creation request where: • The context URI is the URI of the newly-created resource, • The link relation is `http://open-services.net/ns/core#Compact`, and • The target URI is the URI of the associated Compact resource.
rp-11	Clients **MAY** request that the Compact resource is returned inline with the target resource using the `Prefer` request header [RFC7240] and • Preference `return`, value `representation` • Parameter `include` [LDP], value `http://open-services.net/ns/core#PreferCompact`.
rp-12	Servers **MUST** honor a client's request to in-line the Compact resource in JSON and RDF representations if the target resource has a Compact resource and the request is successful.
rp-13	If the target resource does not exist or is not accessible to the requesting client, servers **MUST** return the same status code that it would have returned had the client not included the `Prefer` header in the request.
rp-14	When servers in-line the Compact resource with the target resource in an `application/json` [RFC4627] response, the response entity **MUST** be a JSON object with a `compact` property where the value is the Compact resource JSON as described in Appendix A. JSON Representation Format.
rp-15	Servers **MAY** choose to return only a subset of the target resource properties when a client requests that the Compact resource is returned in-line.
rp-16	Clients **SHOULD** inspect the response body for the Compact resource even if the server omits the `Preference-Applied` header [RFC7240].
rp-17	Servers **MUST** support the `application/json` [RFC4627] and `text/turtle` [turtle] media types for the Compact resource.
rp-18	Servers **SHOULD** support the `application/ld+json` media type [JSON-LD] for the Compact resource.

Clause Number	Requirement
rp-19	The `application/json` Compact resource representation **MUST** be in the JSON format described in Appendix A. JSON Representation Format.
rp-20	Servers **MAY** respond with JSON-LD in compacted document form when clients request the `application/json` representation of the Compact resource as long as the response meets the requirements for the JSON representation.
rp-21	RDF representations of the Compact resource **MUST** conform to the shape specified in 6.1 Resource: Compact.
rp-22	Content-Type `application/x-oslc-compact+xml` representations of the Compact resource **MUST** conform to the shape specified in Appendix B. XML Representation Format.
rp-23	The Compact resource **MAY** have additional server-specific properties.
rp-24	When displaying a preview inside another HTML presentation, clients **MUST** use an `iframe` element, setting its `src` attribute to the preview's document URI.
rp-25	Servers **MUST** express the `oslc:hintWidth` and `oslc:hintHeight` properties of an `oslc:Preview` in length units as specified in [CSS21].
rp-26	Servers **MAY** use dynamic resizing for small and large previews in addition to `oslc:Preview oslc:hintHeight` and `oslc:hintWidth` properties.
rp-27	Servers that support dynamic sizing **MUST** send resize messages to the parent window using the `Window.postMessage` method [webmessaging] where the source of the event is the preview's window.
rp-28	A resize message **MUST** consist of the characters `oslc-resize:` followed by a JSON object with at least one of the following properties: • `oslc:hintHeight` or • `oslc:hintWidth`.
rp-29	A resize message **MAY** also also consist of the characters `oslc-preview-height:` followed by the desired height in pixels.
rp-30	The `oslc:hintHeight` and `oslc:hintWidth` property values in the resize message JSON **MUST** be length units as specified in [CSS21].
rp-31	Servers **MAY** send more than one dynamic resize request.
rp-32	Clients **MAY** choose to ignore the size hints in `oslc:Preview` resources or in dynamic resize messages.
rp-33	An OSLC server providing the Resource Preview capability **MUST** implement the vocabulary defined in this section.

Appendix A. JSON Representation Format

A Compact resource JSON representation might look like the following for the target resource `http://example.com/bugs/324`:

Example 19: Compact Resource JSON

```
{
    "title": "324: Need a fix <em>NOW</em>",
    "shortTitle": "324",
    "icon": "http://example.com/icons/defect.jpg",
    "iconSrcSet": "http://example.com/icons/smallIcon.png 16w, http://example.com/icons/largeIcon.png 64w",
    "iconTitle": "Defect",
    "iconAltLabel": "Defect",
    "largePreview": {
        "document": "http://example.com/bugs/324?preview=large",
        "hintHeight": "250px",
        "hintWidth": "400px"
    },
    "smallPreview": {
        "document": "http://example.com/bugs/324?preview=small"
    }
}
```

The JSON representation has the following constraints:

- The value of the `title` or `shortTitle` properties **MUST**, if present, only contain markup valid in an HTML `span` element. To afford clients the greatest flexibility to style the text to match the context in which it is being displayed, the titles **SHOULD** be plain text with HTML markup used only to emphasize words or phrases or to convey additional information about the target resource.
- `icon`, if present, **MUST** be the URI of an image. The image **SHOULD** be square. A client **MAY** scale the image as needed.
- `iconSrcSet`, if present, **MUST** provide a comma-separated list of image URIs along with image size as specified for the HTML img srcset attribute.
- `iconAltLabel` and `iconTitle`, if present, **MUST** only have string content with no markup.
- `smallPreview` and `largePreview`, if present, **MAY** have any number of properties, including none.
    - The value of `smallPreview` and `largePreview`, if present, **MUST** be the JSON representation of an `oslc:Preview`.
        - A `Preview` **MUST** include a document property. It **MAY** have any number of other properties.
        - `document` **MUST** be the URI of a document containing a preview of the resource.
        - `hintWidth` and `hintHeight`, if present, **MUST** be expressed using length units as specified in [CSS21].

A.1 JSON-LD Context

This section is non-normative.

The following JSON-LD Context may be used with JSON Compact representations.

Example 20: JSON-LD Context for a Compact Resource

```
{
    "_comment": "JSON-LD context for OSLC Resource Preview",
    "@context": {
        "@base": "@https://tools.oasis-open.org/version-control/svn/oslc-core/trunk/specs/contexts/preview.jsonld",
        "oslc" : "http://open-services.net/ns/core#",
        "rdf" : "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
        "dcterms" : "http://purl.org/dc/terms/",
        "@vocab": "http://open-services.net/ns/core#",

        "dcterms:title": {
            "@id": "dcterms:title",
            "@type": "xsd:string"
        },
        "compact": {
            "@id": "oslc:compact",
            "@type": "@id"
        },
        "icon": {
            "@id": "oslc:icon",
            "@type": "@id"
        },
        "iconSrcSet": {
            "@id": "oslc:iconSrcSet",
            "@type": "@id"
```

```
            },
            "iconAltLabel": {
                "@id": "oslc:iconAltLabel",
                "@type": "xsd:string"
            },
            "hintHeight": {
                "@id": "oslc:hintHeight",
                "@type": "xsd:integer"
            },
            "hintWidth": {
                "@id": "oslc:hintWidth",
                "@type": "xsd:integer"
            },
            "shortTitle": {
                "@id": "oslc:iconTitle",
                "@type": "xsd:string"
            },
            "smallPreview": {
                "@id": "oslc:smallPreview",
                "@type": "oslc:Preview"
            },
            "largePreview": {
                "@id": "oslc:largePreview",
                "@type": "oslc:Preview"
            }
        }
    }
}
```

A.2 JSON Schema

This section is non-normative.

The following JSON Schema [JSONSchema] describes the Compact JSON representation.

Example 21: JSON Schema for a Compact Resource

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "title": "OSLC 3.0 Compact JSON",
    "type": "object",
    "properties": {
        "title": {
            "id": "title",
            "type": "string"
        },
        "shortTitle": {
            "id": "shortTitle",
            "type": "string"
        },
        "icon": {
            "id": "icon",
            "type": "string"
        },
        "iconTitle": {
            "id": "iconTitle",
            "type": "string"
        },
        "iconAltLabel": {
            "id": "iconAltLabel",
            "type": "string"
        },
        "largePreview": {
            "id": "largePreview",
            "type": "object",
            "properties": {
                "document": {
                    "id": "document",
                    "type": "string"
                },
                "hintHeight": {
                    "id": "hintHeight",
                    "type": "string",
                    "pattern": "^[0-9]+(\\.[0-9]+)?(em|ex|in|cm|mm|pt|pc|px)$"
                },
                "hintWidth": {
                    "id": "hintWidth",
                    "type": "string",
                    "pattern": "^[0-9]+(\\.[0-9]+)?(em|ex|in|cm|mm|pt|pc|px)$"
                }
            }
        }
    }
}
```

```
            },
            "required": ["document"]
        },
        "smallPreview": {
            "id": "smallPreview",
            "type": "object",
            "properties": {
                "document": {
                    "id": "document",
                    "type": "string"
                },
                "hintHeight": {
                    "id": "hintHeight",
                    "type": "string",
                    "pattern": "^[0-9]+(\\.[0-9]+)?(em|ex|in|cm|mm|pt|pc|px)$"
                },
                "hintWidth": {
                    "id": "hintWidth",
                    "type": "string",
                    "pattern": "^[0-9]+(\\.[0-9]+)?(em|ex|in|cm|mm|pt|pc|px)$"
                }
            },
            "required": ["document"]
        }
    }
}
```

Appendix B. XML Representation Format

A Compact resource XML representation woould look like the following for the target resource `http://example.com/bugs/12345`:

Example 22: Compact Resource XML

```
<?xml version="1.0" encoding="UTF-8"?>

<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:oslc="http://open-services.net/ns/core#">

 <oslc:Compact
    rdf:about="http://example.com/bugs/12345">

    <dcterms:title> 12345: Need a &quot;fix&quot; &lt;em&gt;NOW&lt;/em&gt; </dcterms:title>
    <oslc:shortTitle> 12345 </oslc:shortTitle>
    <oslc:icon rdf:resource="http://example.com/icons/defect.jpg" />
    <oslc:iconAltLabel>Defect</oslc:iconAltLabel>
    <oslc:iconTitle>Defect</oslc:iconTitle>

    <oslc:smallPreview>
       <oslc:Preview>
          <oslc:document rdf:resource="http://example.com/bugs/12345?hover=small" />
       </oslc:Preview>
    </oslc:smallPreview>


    <oslc:largePreview>
       <oslc:Preview>
          <oslc:document rdf:resource="http://example.com/bugs/12345?hover=large" />
          <oslc:hintWidth> 60em </oslc:hintWidth>
          <oslc:hintHeight> 20em </oslc:hintHeight>
       </oslc:Preview>
    </oslc:largePreview>

 </oslc:Compact>

</rdf:RDF>
```

The XML representation has the following syntactic and semantic constraints:

- The root element **MUST** be `rdf:RDF` and **SHOULD** identify the namespaces for RDF (`http://www.w3.org/1999/02/22-rdf-syntax-ns#`), Dublin Core (`http://purl.org/dc/terms/`), and OSLC (`http://open-services.net/ns/core#`), as well as namespaces for any additional Provider supplied properties.
- The `oslc:Compact` element **MUST** be a child of the root element and **MUST** specify an `rdf:about` attribute whose value is the URI of the target resource.
  - The `oslc:Compact` element **MAY** have any number of child elements, including none. Each child element specifies one property of the target resource. The order of child elements is not significant.
  - Each of the `dcterms:title`, `oslc:shortTitle`, `oslc:icon`, `oslc:smallPreview`, and `oslc:largePreview` child elements **MAY** occur at most once.
  - The content of a `dcterms:title` or `oslc:shortTitle` element **MUST** be limited to any XHTML <span> element. To afford clients the greatest flexibility to style the text to match the context in which it is being displayed, the titles **SHOULD** be plain text with XHTML markup used only to emphasize words or phrases or to convey additional information about the target resource.
  - An `oslc:icon` element **MUST** have an `rdf:resource` attribute whose value is a URI of an image. The image **SHOULD** be 16x16 pixels in size.
  - The `oslc:iconAltLabel` and `oslc:iconTitle` elements **MUST** only have string content.
  - The `oslc:smallPreview` and `oslc:largePreview` elements **MAY** have any number of child elements, including none.
    - The `oslc:Preview` element **MUST** be a child of an `oslc:smallPreview` or `oslc:largePreview` element and **MAY** occur at most once.
      - An `oslc:Preview` element **MAY** have any number of child elements, including none. Each child element specifies one property of the Preview. The order of child elements is not significant.
      - Each of the `oslc:document`, `oslc:hintWidth`, `oslc:hintHeight`, and `oslc:initialHeight` child elements **MAY** occur at most once.
      - The `oslc:document` element **MUST** have an `rdf:resource` attribute whose value is a URI of a document containing a preview of the resource.
      - The content of an `oslc:hintWidth`, `oslc:hintHeight`, or `oslc:initialHeight` element **MUST** be a valid relative length value.
- All elements **MAY** have additional atttributes not specified here. All elements, other than `dcterms:title` and `oslc:shortTitle`, **MAY** include

additional child elements not specified here. Consumers **MUST** quietly ignore unknown elements and attributes encountered in a Compact representation.