Network Working Group
Request for Comments: 4437
Category: Experimental

J. Whitehead
U.C. Santa Cruz
G. Clemm
IBM
J. Reschke, Ed.
greenbytes
March 2006

Web Distributed Authoring and Versioning (WebDAV) Redirect Reference Resources

Status of This Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This specification defines an extension to Web Distributed Authoring and Versioning (WebDAV) to allow clients to author HTTP redirect reference resources whose default response is an HTTP/1.1 3xx (Redirection) status code. A redirect reference makes it possible to access the target resourced indirectly through any URI mapped to the redirect reference resource. This specification does not address remapping of trees of resources or regular expression based redirections. There are no integrity guarantees associated with redirect reference resources. Other mechanisms can also be used to achieve the same functionality as this specification. This specification allows operators to experiment with this mechanism and develop experience on what is the best approach to the problem.

Table of Contents

1. Introduction

This specification extends the Web Distributed Authoring Protocol (WebDAV) to enable clients to create new access paths to existing resources. This capability is useful for several reasons.

WebDAV makes it possible to organize HTTP resources into hierarchies, placing them into groupings, known as collections, that are more easily browsed and manipulated than a single flat collection. However, hierarchies require categorization decisions that locate resources at a single location in the hierarchy, a drawback when a resource has multiple valid categories. For example, in a hierarchy of vehicle descriptions containing collections for cars and boats, a description of a combination car/boat vehicle could belong in either collection. Ideally, the description should be accessible from both. Allowing clients to create new URIs that access the existing resource lets them put that resource into multiple collections.

Hierarchies also make resource sharing more difficult, since resources that have utility across many collections are still forced into a single collection. For example, the mathematics department at one university might create a collection of information on fractals that contains bindings to some local resources, but also provides access to some resources at other universities. For many reasons, it may be undesirable to make physical copies of the shared resources: to conserve disk space, to respect copyright constraints, or to make any changes in the shared resources visible automatically. Being able to create new access paths to existing resources in other collections or even on other unrelated systems is useful for this sort of case.

The redirect reference resources defined here provide a mechanism for creating alternative access paths to existing resources. A redirect reference resource is a resource in one collection whose purpose is to redirect requests to another resource (its target), possibly in a different collection. In this way, it allows clients to submit requests to the target resource from another collection. It redirects most requests to the target resource using an HTTP status code from the 3xx range (Redirection), thereby providing a form of mediated access to the target resource.

A redirect reference is a resource with properties but with no body of its own. Properties of a redirect reference resource can contain information such as who created the reference, when, and why. Since redirect reference resources are implemented using HTTP 3xx responses, it generally takes two round trips to submit a request to the intended resource. Redirect references work equally well for local resources and for resources that reside on a different system from the reference.

The remainder of this document is structured as follows: Section 3 defines terms that will be used throughout the specification. Section 4 provides an overview of redirect reference resources. Section 5 defines the semantics of existing methods when applied to redirect reference resources. Section 6 discusses how to create a redirect reference resource, and Section 7 discusses updating redirect references. Section 8 discusses their semantics when applied to collections that contain redirect reference resources. Sections 9 through 11 discuss several other issues raised by the existence of redirect reference resources. Sections 12 through 15 define the new headers, properties, and XML elements required to support redirect reference resources. Section 16 discusses capability discovery. Sections 17 through 19 present the security, internationalization, and IANA concerns raised by this specification. The remaining sections provide a variety of supporting information.

2. Notational Conventions

Since this document describes a set of extensions to the WebDAV Distributed Authoring Protocol [RFC2518], itself an extension to the HTTP/1.1 protocol, the augmented BNF used here to describe protocol elements is exactly the same as described in Section 2.1 of [RFC2616]. Since this augmented BNF uses the basic production rules provided in Section 2.2 of [RFC2616], these rules apply to this document as well.

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC2119].

3. Terminology

The terminology used here follows and extends that in the WebDAV Distributed Authoring Protocol specification [RFC2518]. Definitions of the terms resource, Uniform Resource Identifier (URI), and Uniform Resource Locator (URL) are provided in [RFC3986].

Redirect Reference Resource

A resource created to redirect all requests made to it, using an HTTP status code from the 3xx range, to a defined target resource.

Non-Reference Resource

A resource that is not a reference to another resource.

Target Resource

The resource to which requests are redirected by a redirect reference resource. A target resource can be anything that can be identified by an absolute URI (see [RFC3986], "absolute-URI").

This document uses the terms "precondition", "postcondition", and "protected property" as defined in [RFC3253]. Servers MUST report pre-/postcondition failures as described in Section 1.6 of this document.

4. Overview of Redirect Reference Resources

For all operations submitted to a redirect reference resource, the default response is a 302 (Found), accompanied by the Redirect-Ref header (defined in Section 12.1, below) and the Location header ([RFC2616], Section 14.30) set to the URI of the target resource. With this information, the client can resubmit the request to the URI of the target resource.

A redirect reference resource never automatically forwards requests to its target resource. Redirect resources bring the same benefits as links in HTML documents. They can be created and maintained without the involvement or even knowledge of their target resource. This reduces the cost of linking between resources.

If the client is aware that it is operating on a redirect reference resource, it can resolve the reference by retrieving the reference resource's DAV:reftarget property (defined in Section 13.2, below), whose value contains the URI of the target resource. It can then submit requests to the target resource.

A redirect reference resource is a new type of resource. To distinguish redirect reference resources from non-reference resources, a new value of the DAV:resourcetype property (defined in [RFC2518]), DAV:redirectref, is defined in Section 14.1, below.

Since a redirect reference resource is a resource, methods can be applied to the reference resource as well as to its target resource.

The Apply-To-Redirect-Ref request header (defined in Section 12.2, below) is provided so that referencing-aware clients can control whether an operation is applied to the redirect reference resource or standard HTTP/WebDAV behaviour (redirection with a 3xx status code) should occur. The Apply-To-Redirect-Ref header can be used with most requests to redirect reference resources. This header is particularly useful with PROPFIND, to retrieve the reference resource's own properties.

Implementation Note: Operations on the target of a redirect reference usually do not affect the redirect reference itself. However, clients should not rely on this behaviour (for instance, some servers may update redirect references as a result of namespace operations on the reference's target).

5. Operations on Redirect Reference Resources

Although non-referencing-aware clients cannot create reference resources, they should be able to submit requests through the reference resources created by reference-aware WebDAV clients. They should be able to follow any references to their targets. To make this possible, a server that receives any request made via a redirect reference resource MUST return a 3xx range (Redirection) status code, unless the request includes an Apply-To-Redirect-Ref header specifying "T". The client and server MUST follow [RFC2616], Section 10.3, but with these additional rules:

- The Location response header MUST contain a URI (see [RFC3986], Section 3) that identifies the target of the reference resource.

- The response MUST include the Redirect-Ref header. This header allows reference-aware WebDAV clients to recognize the resource as a reference resource and to understand the reason for the redirection.

A reference-aware WebDAV client can, like a non-referencing client, resubmit the request to the URI in the Location header in order to operate on the target resource. Alternatively, it can resubmit the request to the URI of the redirect reference resource with the "Apply-To-Redirect-Ref: T" header in order to operate on the reference resource itself. In this case, the request MUST be applied to the reference resource itself, and a 3xx response MUST NOT be returned.

As redirect references do not have bodies, GET and PUT requests with "Apply-To-Redirect-Ref: T" MUST fail with status 403 (forbidden).

6. MKREDIRECTREF Method

The MKREDIRECTREF method requests the creation of a redirect reference resource.

If a MKREDIRECTREF request fails, the server state preceding the request MUST be restored.

Responses from a MKREDIRECTREF request MUST NOT be cached, as MKREDIRECTREF has non-idempotent and non-safe semantics (see [RFC2616], Section 9.1).

Marshalling

The request body MUST be a DAV:mkredirectref XML element.

```
<!ELEMENT mkredirectref (reftarget, redirect-lifetime?)>
<!ELEMENT reftarget (href)>
<!ELEMENT redirect-lifetime (permanent | temporary)>
<!ELEMENT permanent EMPTY>
<!ELEMENT temporary EMPTY>
```

The DAV:href element is defined in [RFC2518] (Section 12.3) and MUST contain either a URI or a relative-ref (see [RFC3986], Sections 3 and 4.2).

If no DAV:redirect-lifetime element is specified, the server MUST behave as if a value of DAV:temporary was specified.

If the request succeeds, the server MUST return 201 (Created) status.

If a response body for a successful request is included, it MUST be a DAV:mkredirectref-response XML element. Note that this document does not define any elements for the MKREDIRECTREF response body, but the DAV:mkredirectref-response element is defined to ensure interoperability between future extensions that do define elements for the response body.

```
<!ELEMENT mkredirectref-response ANY>
```

Preconditions

(DAV:resource-must-be-null): A resource MUST NOT exist at the Request-URI.

(DAV:parent-resource-must-be-non-null): The Request-URI minus the last past segment MUST identify a collection.

(DAV:name-allowed): The last segment of the Request-URI is available for use as a resource name.

(DAV:locked-update-allowed): If the collection identified by the Request-URI minus the last path segment is write-locked, then the appropriate token MUST be specified in an If request header.

(DAV:redirect-lifetime-supported): If the request body contains a DAV:redirect-lifetime element, the server MUST support the specified lifetime. Support for DAV:temporary is REQUIRED, while support for DAV:permanent is OPTIONAL.

(DAV:legal-reftarget): The specified is a legal URI or relative-ref.

Postconditions

(DAV:new-redirectref): a new redirect reference resource is created whose DAV:reftarget property has the value specified in the request body.

6.1. Example: Creating a Redirect Reference Resource with MKREDIRECTREF

>> Request:

```
MKREDIRECTREF /~whitehead/dav/spec08.ref HTTP/1.1
Host: www.example.com
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx

<?xml version="1.0" encoding="utf-8" ?>
<D:mkredirectref xmlns:D="DAV:">
  <D:reftarget>
    <D:href>/i-d/draft-webdav-protocol-08.txt</D:href>
  </D:reftarget>
</D:mkredirectref>
```

>> Response:

```
HTTP/1.1 201 Created
```

This request resulted in the creation of a new redirect reference resource at http://www.example.com/~whitehead/dav/spec08.ref, which points to the resource identified by the DAV:reftarget property. In this example, the target resource is identified by the URI http://www.example.com/i-d/draft-webdav-protocol-08.txt. The redirect reference resource's DAV:resourcetype property is set to DAV:redirectref, and its DAV:redirect-lifetime property has the value DAV:temporary.

7. UPDATEREDIRECTREF Method

The UPDATEREDIRECTREF method requests the update of a redirect reference resource.

If a UPDATEREDIRECTREF request fails, the server state preceding the request MUST be restored.

Responses from a UPDATEREDIRECTREF request MUST NOT be cached, as UPDATEREDIRECTREF has non-safe semantics (see [RFC2616], Section 9.1).

Marshalling

The request body MUST be a DAV:updateredirectref XML element.

```
<!ELEMENT updateredirectref (reftarget?, redirect-lifetime?)>
```

See Section 6 for a definition of DAV:reftarget and DAV:redirect-lifetime.

If no DAV:reftarget element is specified, the server MUST NOT change the target of the redirect reference.

If no DAV:redirect-lifetime element is specified, the server MUST NOT change the lifetime of the redirect reference.

If a response body for a successful request is included, it MUST be a DAV:updateredirectref-response XML element. Note that this document does not define any elements for the UPDATEREDIRECTREF response body, but the DAV:updateredirectref-response element is defined to ensure interoperability between future extensions that do define elements for the response body.

```
<!ELEMENT updateredirectref-response ANY>
```

Preconditions

(DAV:locked-update-allowed): if the resource is write-locked, then the appropriate token MUST be specified in an If request header.

(DAV:must-be-redirectref): the resource identified by the Request-URI must be a redirect reference resource as defined by this specification.

(DAV:redirect-lifetime-supported): see Section 6.

(DAV:redirect-lifetime-update-supported): servers MAY support changing the DAV:redirect-lifetime property; if they don't, this condition code can be used to signal failure.

(DAV:legal-reftarget): see Section 6.

Postconditions

(DAV:redirectref-updated): the DAV:reftarget and DAV:redirect-lifetime properties of the redirect reference have been updated accordingly.

7.1. Example: Updating a Redirect Reference Resource with UPDATEREDIRECTREF

>> Request:

```
UPDATEREDIRECTREF /~whitehead/dav/spec08.ref HTTP/1.1
Host: www.example.com
Apply-To-Redirect-Ref: T
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx

<?xml version="1.0" encoding="utf-8" ?>
<D:updateredirectref xmlns:D="DAV:">
  <D:reftarget>
    <D:href>/i-d/draft-webdav-protocol-08b.txt</D:href>
  </D:reftarget>
</D:updateredirectref>
```

>> Response:

```
HTTP/1.1 200 OK
```

This request has updated the redirect reference's DAV:reftarget property to "/i-d/draft-webdav-protocol-08b.txt" and has not changed the DAV:redirect-lifetime value. Note that the "Apply-To-Redirect-

Ref" request header must be used; otherwise, the request would result in a redirect (3xx) response status.

8. Operations on Collections That Contain Redirect Reference Resources

According to [RFC2518], Section 9.2, methods that have defined interactions with the "Depth" request header should apply all other request headers to each resource in scope. However, applying this principle to the "Apply-To-Redirect-Ref" header uniformly would make it impractical to implement this specification on top of existing servers and also would result in unexpected server behaviour for clients that do not take the existence of redirect references into account. On the other hand, the definition of the "Depth" header allows alternate behaviours to be explicitly defined.

For this reason, this specification defines the interaction between "Depth" and "Apply-To-Redirect-Ref" request headers on a case-by-case basis and also provides a default for methods not mentioned here that do not specify the behaviour themselves.

method name	defined in	supported depths	behaviour
COPY	[RFC2518], 8.9	0, infinity	"T"
DELETE	[RFC2518], 8.7	infinity	"T"
LOCK	[RFC2518], 8.11	0, infinity	"T"
MOVE	[RFC2518], 8.10	0, infinity	"T"
PROPFIND	[RFC2518], 8.2	0, 1, infinity	inherit
REPORT	[RFC3253], 3.6	0, 1, infinity	inherit
default			"T"

When the behaviour is defined to be "inherit", the method should follow RFC2518's default behaviour for "Depth" operations, which means applying the value given for "Apply-To-Redirect-Ref" to each resource in scope. On the other hand, when it is defined to be "T", the method should behave as if a "Apply-To-Redirect-Ref: T" header was specified for each operation on child resources. The latter ensures that "Depth: infinity" operations will not fail unexpectedly just because there was a redirect reference resource in scope.

8.1. Example: PROPFIND on a Collection with Redirect Reference Resources

Suppose a PROPFIND request with Depth: infinity is submitted to the following collection, with the members shown here:

```
   /MyCollection/
        (non-reference resource) diary.html
        (redirect reference resource) nunavut

   >> Request:

   PROPFIND /MyCollection/ HTTP/1.1
   Host: example.com
   Depth: infinity
   Apply-To-Redirect-Ref: F
   Content-Type: text/xml
   Content-Length: xxxx

   <?xml version="1.0" ?>
   <D:propfind xmlns:D="DAV: ">
     <D:prop xmlns:J="http://example.com/jsprops/">
       <D:resourcetype/>
       <J:keywords/>
     </D:prop>
   </D:propfind>

   >> Response:

   HTTP/1.1 207 Multi-Status
   Content-Type: text/xml
   Content-Length: xxxx

   <?xml version="1.0" ?>
   <D:multistatus xmlns:D="DAV:" xmlns:J="http://example.com/jsprops/">
     <D:response>
       <D:href>/MyCollection/</D:href>
       <D:propstat>
         <D:prop>
           <D:resourcetype><D:collection/></D:resourcetype>
           <J:keywords>diary, interests, hobbies</J:keywords>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
     <D:response>
       <D:href>/MyCollection/diary.html</D:href>
       <D:propstat>
         <D:prop>
           <D:resourcetype/>
           <J:keywords>diary, travel, family, history</J:keywords>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
```

```
    </D:response>
    <D:response>
      <D:href>/MyCollection/nunavut</D:href>
      <D:status>HTTP/1.1 302 Found</D:status>
      <D:location>
        <D:href>http://example.ca/art/inuit/</D:href>
      </D:location>
    </D:response>
  </D:multistatus>
```

In this example, the Depth header is set to infinity, and the Apply-To-Redirect-Ref header is set to "F". The collection contains one URI that identifies a redirect reference resource. The response element for the redirect reference resource has a status of 302 (Found) and includes a DAV:location extension element to allow clients to retrieve the properties of its target resource. (The response element for the redirect reference resource does not include the requested properties. The client can submit another PROPFIND request to the URI in the DAV:location pseudo-property to retrieve those properties.)

8.2. Example: PROPFIND with Apply-To-Redirect-Ref on a Collection with Redirect Reference Resources

Suppose a PROPFIND request with "Apply-To-Redirect-Ref: T" and Depth: infinity is submitted to the following collection, with the members shown here:

```
/MyCollection/
     (non-reference resource) diary.html
     (redirect reference resource) nunavut
```

>> Request:

```
PROPFIND /MyCollection/ HTTP/1.1
Host: example.com
Depth: infinity
Apply-To-Redirect-Ref: T
Content-Type: text/xml
Content-Length: xxxx
```

```
<?xml version="1.0" ?>
<D:propfind xmlns:D="DAV:">
  <D:prop>
    <D:resourcetype/>
    <D:reftarget/>
    <D:redirect-lifetime/>
  </D:prop>
</D:propfind>

>> Response:

HTTP/1.1 207 Multi-Status
Content-Type: text/xml
Content-Length: xxxx

<?xml version="1.0" ?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>/MyCollection/</D:href>
    <D:propstat>
      <D:prop>
        <D:resourcetype><D:collection/></D:resourcetype>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
    <D:propstat>
      <D:prop>
        <D:reftarget/>
        <D:redirect-lifetime/>
      </D:prop>
      <D:status>HTTP/1.1 404 Not Found</D:status>
    </D:propstat>
  </D:response>
  <D:response>
    <D:href>/MyCollection/diary.html</D:href>
    <D:propstat>
      <D:prop>
        <D:resourcetype/>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
    <D:propstat>
      <D:prop>
        <D:reftarget/>
        <D:redirect-lifetime/>
      </D:prop>
      <D:status>HTTP/1.1 404 Not Found</D:status>
    </D:propstat>
```

```
    </D:response>
    <D:response>
      <D:href>/MyCollection/nunavut</D:href>
      <D:propstat>
        <D:prop>
          <D:resourcetype><D:redirectref/></D:resourcetype>
          <D:reftarget>
            <D:href>http://example.ca/art/inuit/</D:href>
          </D:reftarget>
          <D:redirect-lifetime><D:temporary/></D:redirect-lifetime>
        </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
      </D:propstat>
    </D:response>
  </D:multistatus>
```

Since the "Apply-To-Redirect-Ref: T" header is present, the response shows the properties of the redirect reference resource in the collection rather than reporting a 302 status.

9. Operations on Targets of Redirect Reference Resources

Operations on targets of redirect reference resources have no effect on the reference resource.

10. Relative References in DAV:reftarget

The URI in the href in a DAV:reftarget property MAY be a relative reference. In this case, the base URI to be used for resolving it to absolute form is the URI used in the HTTP message to identify the redirect reference resource to which the DAV:reftarget property belongs.

When DAV:reftarget appears in the context of a Multi-Status response, it is in a DAV:response element that contains a single DAV:href element. The value of this DAV:href element serves as the base URI for resolving a relative reference in DAV:reftarget. The value of DAV:href may itself be relative, in which case it must be resolved first in order to serve as the base URI for the relative reference in DAV:reftarget. If the DAV:href element is relative, its base URI is constructed from the scheme component "http", the value of the Host header in the request, and the Request-URI.

10.1. Example: Resolving a Relative Reference in a Multi-Status Response

```
>> Request:

PROPFIND /geog/ HTTP/1.1
Host: example.com
Apply-To-Redirect-Ref: T
Depth: 1
Content-Type: text/xml
Content-Length: nnn

<?xml version="1.0" ?>
<D:propfind xmlns:D="DAV:">
  <D:prop>
    <D:resourcetype/>
    <D:reftarget/>
  </D:prop>
</D:propfind>

>> Response:

HTTP/1.1 207 Multi-Status
Content-Type: text/xml
Content-Length: nnn

<?xml version="1/0" ?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>/geog/</D:href>
    <D:propstat>
      <D:prop>
        <D:resourcetype><D:collection/></D:resourcetype>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
    <D:propstat>
      <D:prop><D:reftarget/></D:prop>
      <D:status>HTTP/1.1 404 Not Found</D:status>
    </D:propstat>
  </D:response>
  <D:response>
    <D:href>/geog/stats.html</D:href>
    <D:propstat>
      <D:prop>
        <D:resourcetype><D:redirectref/></D:resourcetype>
        <D:reftarget>
          <D:href>statistics/population/1997.html</D:href>
```

```
          </D:reftarget>
        </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
      </D:propstat>
    </D:response>
  </D:multistatus>
```

In this example, the relative reference "statistics/population/1997.html" is returned as the value of the DAV:reftarget property for the reference resource identified by href /geog/stats.html. The href is itself a relative reference, which resolves to http://example.com/geog/stats.html. This is the base URI for resolving the relative reference in reftarget. The absolute URI of reftarget is http://example.com/geog/statistics/population/1997.html.

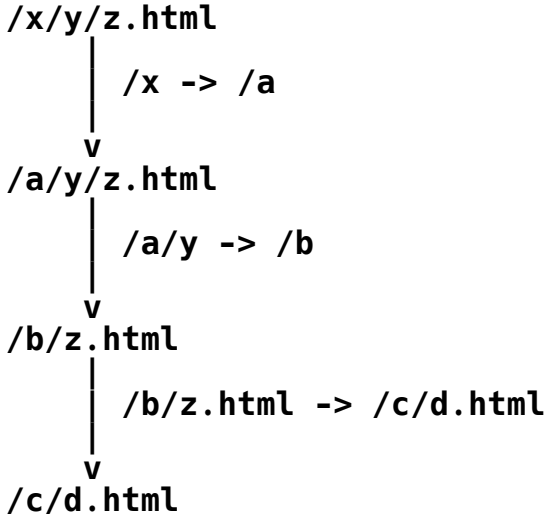
11. Redirect References to Collections

In a Request-URI /segment1/segment2/segment3, any of the three segments may identify a redirect reference resource. (See [RFC3986], Section 3.3, for definitions of "path" and "segment".) If any segment in a Request-URI identifies a redirect reference resource, the response SHOULD be a 3xx. The value of the Location header in the response is as follows:

The leftmost path segment of the Request-URI that identifies a redirect reference resource, together with all path segments and separators to the left of it, is replaced by the value of the redirect reference resource's DAV:reftarget property (resolved to an absolute URI). The remainder of the Request-URI is concatenated to this path.

Note: If the DAV:reftarget property ends with a "/" and the remainder of the Request-URI is non-empty (and therefore must begin with a "/"), the final "/" in the DAV:reftarget property is dropped before the remainder of the Request-URI is appended.

Consider Request-URI /x/y/z.html. Suppose that /x/ is a redirect reference resource, whose target resource is collection /a/, which contains redirect reference resource y whose target resource is collection /b/, which contains redirect reference resource z.html, whose target resource is /c/d.html.



In this case, the client must follow up three separate 3xx responses before finally reaching the target resource. The server responds to the initial request with a 3xx with Location: /a/y/z.html, and the client resubmits the request to /a/y/z.html. The server responds to this request with a 3xx with Location: /b/z.html, and the client resubmits the request to /b/z.html. The server responds to this request with a 3xx with Location: /c/d.html, and the client resubmits the request to /c/d.html. This final request succeeds.

> Note: The behaviour described above may have a very serious impact on the efficiency of mapping Request-URIs to resources in HTTP request processing. Therefore, servers MAY respond with a 404 status code if the cost of checking all leading path segments for redirect references seems prohibitive.

12. Headers

12.1. Redirect-Ref Response Header

```
Redirect-Ref = "Redirect-Ref:" (URI | relative-ref)
; URI: see [RFC3986], Section 3
; relative-ref: see [RFC3986], Section 4.2
```

The Redirect-Ref header is used in all 3xx responses from redirect reference resources. The value is the link target as specified during redirect reference resource creation.

12.2. Apply-To-Redirect-Ref Request Header

```
Apply-To-Redirect-Ref = "Apply-To-Redirect-Ref" ":" ("T" | "F")
```

The optional Apply-To-Redirect-Ref header can be used on any request to a redirect reference resource. When it is present and set to "T", the request MUST be applied to the reference resource itself, and a 3xx response MUST NOT be returned.

If the Apply-To-Redirect-Ref header is used on a request to any other sort of resource besides a redirect reference resource, the server MUST ignore it.

13. Redirect Reference Resource Properties

The properties defined below are REQUIRED on redirect reference resources. A PROPFIND/allprop request SHOULD NOT return any of the properties defined in this document.

13.1. DAV:redirect-lifetime (protected)

This property provides information about the lifetime of a redirect. It can be either DAV:permanent (HTTP status 301) or DAV:temporary (HTTP status 302). Future protocols may define additional values.

```
<!ELEMENT redirect-lifetime (permanent | temporary)>
<!ELEMENT permanent EMPTY>
<!ELEMENT temporary EMPTY>
```

13.2. DAV:reftarget (protected)

This property provides an efficient way for clients to discover the URI of the target resource. This is a read-only property after its initial creation. Its value can only be set in a MKREDIRECTREF request. The value is a DAV:href element containing the URI of the target resource.

```
<!ELEMENT reftarget href >
```

14. XML Elements

14.1. redirectref XML Element

Name: redirectref

Namespace: DAV:

Purpose: Used as the value of the DAV:resourcetype property to specify that the resource type is a redirect reference resource.

```
<!ELEMENT redirectref EMPTY >
```

15. Extensions to the DAV:response XML Element for Multi-Status Responses

As described in Section 8, the DAV:location element may be returned in the DAV:response element of a 207 Multi-Status response, to allow clients to resubmit their requests to the target resource of a redirect reference resource.

Consequently, the definition of the DAV:response XML element changes to the following:

```
<!ELEMENT response (href, ((href*, status)|(propstat+)),
                    responsedescription?, location?) >
<!ELEMENT location (href) >
```

16. Capability Discovery

Sections 9.1 and 15 of [RFC2518] describe the use of compliance classes with the DAV header in responses to OPTIONS, to indicate which parts of the WebDAV Distributed Authoring protocols the resource supports. This specification defines an OPTIONAL extension to [RFC2518]. It defines a new compliance class, called redirectrefs, for use with the DAV header in responses to OPTIONS requests. If a resource does support redirect references, its response to an OPTIONS request may indicate that it does, by listing the new redirectrefs compliance class in the DAV header and by listing the MKREDIRECTREF method as one it supports.

When responding to an OPTIONS request, any type of resource can include redirectrefs in the value of the DAV header. Doing so indicates that the server permits a redirect reference resource at the Request-URI.

16.1. Example: Discovery of Support for Redirect Reference Resources

>> Request:

```
OPTIONS /somecollection/someresource HTTP/1.1
Host: example.org
```

```
>> Response:

HTTP/1.1 200 OK
Allow: OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, COPY, MOVE
Allow: MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, MKREDIRECTREF
DAV: 1, 2, redirectrefs
```

The DAV header in the response indicates that the resource /somecollection/someresource is level 1 and level 2 compliant, as defined in [RFC2518]. In addition, /somecollection/someresource supports redirect reference resources. The Allow header indicates that MKREDIRECTREF requests can be submitted to /somecollection/someresource.

17. Security Considerations

This section is provided to make applications that implement this protocol aware of the security implications of this protocol.

All of the security considerations of HTTP/1.1 and the WebDAV Distributed Authoring Protocol specification also apply to this protocol specification. In addition, redirect reference resources introduce several new security concerns and increase the risk of some existing threats. These issues are detailed below.

17.1. Privacy Concerns

By creating redirect reference resources on a trusted server, it is possible for a hostile agent to induce users to send private information to a target on an unrelated system. This risk is mitigated somewhat, since clients are required to notify the user of the redirection for any request other than GET or HEAD. (See [RFC2616], Section 10.3.3, 302 Found.)

17.2. Redirect Loops

Although redirect loops were already possible in HTTP 1.1, the introduction of the MKREDIRECTREF method creates a new avenue for clients to create loops accidentally or maliciously. If the reference resource and its target are on the same server, the server may be able to detect MKREDIRECTREF requests that would create loops. See also [RFC2616], Section 10.3, "Redirection 3xx."

17.3. Redirect Reference Resources and Denial of Service

Denial of service attacks were already possible by posting URLs that were intended for limited use at heavily used Web sites. The introduction of MKREDIRECTREF creates a new avenue for similar denial

of service attacks. Clients can now create redirect reference resources at heavily used sites to target locations that were not designed for heavy usage.

17.4. Revealing Private Locations

There are several ways that redirect reference resources may reveal information about collection structures. First, the DAV:reftarget property of every redirect reference resource contains the URI of the target resource. Anyone who has access to the reference resource can discover the collection path that leads to the target resource. The owner of the target resource may have wanted to limit knowledge of this collection structure.

Sufficiently powerful access control mechanisms can control this risk to some extent. Property-level access control could prevent users from examining the DAV:reftarget property. (The Location header returned in responses to requests on redirect reference resources reveals the same information, however.)

This risk is no greater than the similar risk posed by HTML links.

18. Internationalization Considerations

All internationalization considerations mentioned in [RFC2518] also apply to this document.

19. IANA Considerations

All IANA considerations mentioned in [RFC2518] also apply to this document.

19.1. HTTP headers

This document specifies the two new HTTP headers listed below.

19.1.1. Redirect-Ref

Header field name: Redirect-Ref

Applicable protocol: http

Status: standard

Author/Change controller: IETF

Specification document: this specification (Section 12.1)

19.1.2 Apply-To-Redirect-Ref

Header field name: Apply-To-Redirect-Ref

Applicable protocol: http

Status: standard

Author/Change controller: IETF

Specification document: this specification (Section 12.2)

20. Contributors

Many thanks to Jason Crawford, Jim Davis, Chuck Fay, and Judith Slein, who can take credit for big parts of the original design of this specification.

21. Acknowledgements

This document has benefited from thoughtful discussion by Jim Amsden, Peter Carlson, Steve Carter, Tyson Chihaya, Ken Coar, Ellis Cohen, Bruce Cragun, Spencer Dawkins, Mark Day, Rajiv Dulepet, David Durand, Lisa Dusseault, Stefan Eissing, Roy Fielding, Yaron Goland, Fred Hitt, Alex Hopmann, James Hunt, Marcus Jager, Chris Kaler, Manoj Kasichainula, Rohit Khare, Daniel LaLiberte, Steve Martin, Larry Masinter, Jeff McAffer, Joe Orton, Surendra Koduru Reddy, Juergen Reuter, Max Rible, Sam Ruby, Bradley Sergeant, Nick Shelness, John Stracke, John Tigue, John Turner, Kevin Wiggen, and others.

22. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2518] Goland, Y., Whitehead, E., Faizi, A., Carter, S., and D. Jensen, "HTTP Extensions for Distributed Authoring -- WEBDAV", RFC 2518, February 1999.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3253] Clemm, G., Amsden, J., Ellison, T., Kaler, C., and J. Whitehead, "Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning)", RFC 3253, March 2002.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

Authors' Addresses

Jim Whitehead
UC Santa Cruz, Dept. of Computer Science
1156 High Street
Santa Cruz, CA 95064
US

EMail: ejw@cse.ucsc.edu

Geoff Clemm
IBM
20 Maguire Road
Lexington, MA 02421
US

EMail: geoffrey.clemm@us.ibm.com

Julian F. Reschke (editor)
greenbytes GmbH
Hafenweg 16
Muenster, NW 48155
Germany

Phone: +49 251 2807760
Fax: +49 251 2807761
EMail: julian.reschke@greenbytes.de
URI: http://greenbytes.de/tech/webdav/

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights 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; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat 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 specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).