

Independent Submission

K. Murchison

Internet-Draft

CMU

Updates: [3253](#), [4791](#), [4918](#), [5689](#), [6352](#),
[6638](#) (if approved)

January 16, 2013

Intended status: Standards Track

Expires: July 20, 2013

**Use of the Prefer Header Field in Web Distributed Authoring and
Versioning (WebDAV)
draft-murchison-webdav-prefer-01**

Abstract

This specification defines how the HTTP Prefer header can be used by a WebDAV client to request that certain behaviors be employed by a server while constructing a response to a successful request.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 20, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Notational Conventions	3
2.	Reducing WebDAV Response Verbosity with "return=minimal"	3
2.1.	Minimal PROPFIND Response	3
2.1.1.	Example: Typical PROPFIND request/response	4
2.1.2.	Example: Minimal PROPFIND request/response	5
2.1.3.	Example: Minimal PROPFIND request/response with an empty DAV:propstat element	7
2.2.	Minimal REPORT Response	8
2.2.1.	Example: Typical REPORT request/response	8
2.2.2.	Example: Minimal REPORT request/response	10
2.3.	Minimal PROPPATCH Response	11
2.3.1.	Example: Typical PROPPATCH request/response	12
2.3.2.	Example: Minimal PROPPATCH request/response	13
2.4.	Minimal MKCALENDAR / MKCOL Response	13
2.4.1.	Example: Verbose MKCOL request/response	14
2.4.2.	Example: Minimal MKCOL request/response	15
3.	Reducing WebDAV Round-Trips with "return=representation"	15
3.1.	Example: Typical resource creation and retrieval via POST + GET	16
3.2.	Example: Streamlined resource creation and retrieval via POST	18
4.	The "depth-noroot" Processing Preference	19
4.1.	Example: Typical PROPFIND request/response with Depth:1	20
4.2.	Example: PROPFIND request/response with Depth:1 and Prefer:depth-noroot	21
5.	Security Considerations	23
6.	IANA Considerations	23
7.	Acknowledgements	23
8.	References	23
8.1.	Normative References	23
8.2.	Informative References	24
Appendix A.	The Brief and Extended Depth Request Header Fields	25
Appendix B.	Open Issues	25
Appendix C.	Change Log (to be removed by RFC Editor before publication)	25
C.1.	Since -00	25
C.2.	Since CalConnect XXIV	25
	Author's Address	26

1. Introduction

[I-D.snell-http-prefer] defines the HTTP Prefer request header field and the "return=minimal" preference which indicates that a client wishes for the server to return a minimal response to a successful request, but states that what constitutes an appropriate minimal response is left solely to the discretion of the server. [Section 2](#) of this specification defines precisely what is expected of a server when constructing minimal responses to successful WebDAV [\[RFC4918\]](#) requests.

[I-D.snell-http-prefer] also defines the "return=representaion" preference which indicates that a client wishes for the server to include an entity representing the current state of the resource in the response to a successful request. The behavior of this preference with WebDAV [\[RFC4918\]](#) requests needs no further clarification, but [Section 3](#) of this specification makes recommendations on when it should be used by clients.

Finally, [Section 4](#) of this specifcation defines the "depth-noroot" preference that can be used with WebDAV [\[RFC4918\]](#) methods that support the "Depth" header field..

1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

This document references XML elements types in the "DAV:" namespace outside of the context of an XML fragment. When doing so, the string "DAV:" will be prepended to the XML element type.

2. Reducing WebDAV Response Verbosity with "return=minimal"

Some payload bodies in responses to WebDAV [\[RFC4918\]](#) requests, such as 207 (Multi-Status) [\[RFC4918\]](#) responses, can be quite verbose or even unnecessary at times. This specification defines how the Prefer [\[I-D.snell-http-prefer\]](#) request header field, in conjunction with its "return=minimal" preference, can be used by clients to reduce the verbosity of such responses by requesting that the server omit those portions of the response that can be inferred by their absence.

2.1. Minimal PROPFIND Response

When a PROPFIND [\[RFC4918\]](#) method request contains a Prefer [\[I-D.snell-http-prefer\]](#) header field with a preference of

"return=minimal", the server SHOULD omit all DAV:propstat XML elements containing a DAV:status XML element of value 404 (Not Found) [[I-D.ietf-httpbis-p2-semantics](#)] from the 207 (Multi-Status) [[RFC4918](#)] response. If the omission of such a DAV:propstat element would result in a DAV:response XML element containing zero DAV:propstat elements, then the server MUST substitute a DAV:propstat element consisting of an empty DAV:prop element and a DAV:status element of value 200 (OK) [[I-D.ietf-httpbis-p2-semantics](#)] in its place.

2.1.1. Example: Typical PROPFIND request/response

This example tries to fetch an unknown property from a CARDDAV: addressbook [[RFC6352](#)] collection.

>> Request <<

PROPFIND /container/ HTTP/1.1

Host: webdav.example.com

Content-Type: application/xml; charset=utf-8

Content-Length: xxxx

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

```
<D:propfind xmlns:D="DAV:" xmlns:X="http://ns.example.com/foobar/">
```

```
  <D:prop>
```

```
    <D:resourcetype/>
```

```
    <X:foobar/>
```

```
  </D:prop>
```

```
</D:propfind>
```


>> Response <<

HTTP/1.1 207 Multi-Status

Content-Type: application/xml; charset=utf-8

Content-Length: xxxx

Vary: Prefer

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

```
<D:multistatus xmlns:D="DAV:" xmlns:X="http://ns.example.com/foobar/"
  xmlns:C="urn:ietf:params:xml:ns:carddav">
```

```
  <D:response>
```

```
    <D:href>/container/</D:href>
```

```
    <D:propstat>
```

```
      <D:prop>
```

```
        <D:resourcetype>
```

```
          <D:collection/>
```

```
          <C:addressbook/>
```

```
        </D:resourcetype>
```

```
      </D:prop>
```

```
      <D:status>HTTP/1.1 200 OK</D:status>
```

```
    </D:propstat>
```

```
    <D:propstat>
```

```
      <D:prop>
```

```
        <X:foobar/>
```

```
      </D:prop>
```

```
      <D:status>HTTP/1.1 404 Not Found</D:status>
```

```
    </D:propstat>
```

```
  </D:response>
```

```
</D:multistatus>
```

2.1.2. Example: Minimal PROPFIND request/response

This example tries to fetch an unknown property from a CARDDAV:
addressbook [[RFC6352](#)] collection.

>> Request <<

```
PROPFIND /container/ HTTP/1.1
Host: webdav.example.com
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx
Prefer: return=minimal
```

```
<?xml version="1.0" encoding="UTF-8"?>
<D:propfind xmlns:D="DAV:" xmlns:X="http://ns.example.com/foobar/">
  <D:prop>
    <D:resourcetype/>
    <X:foobar/>
  </D:prop>
</D:propfind>
```

>> Response <<

```
HTTP/1.1 207 Multi-Status
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx
Preference-Applied: return=minimal
Vary: Prefer
```

```
<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:" xmlns:C="urn:ietf:params:xml:ns:carddav">
  <D:response>
    <D:href>/container/</D:href>
    <D:propstat>
      <D:prop>
        <D:resourcetype>
          <D:collection/>
          <C:addressbook/>
        </D:resourcetype>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```


2.1.3. Example: Minimal PROPFIND request/response with an empty DAV: propstat element

This example tries to fetch an unknown property from a CARDDAV: addressbook [[RFC6352](#)] collection.

>> Request <<

```
PROPFIND /container/ HTTP/1.1
Host: webdav.example.com
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx
Prefer: return=minimal

<?xml version="1.0" encoding="UTF-8"?>
<D:propfind xmlns:D="DAV:" xmlns:X="http://ns.example.com/foobar/">
  <D:prop>
    <X:foobar/>
  </D:prop>
</D:propfind>
```

>> Response <<

```
HTTP/1.1 207 Multi-Status
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx
Preference-Applied: return=minimal
Vary: Prefer

<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>/container/</D:href>
    <D:propstat>
      <D:prop/>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```


2.2. Minimal REPORT Response

When a REPORT method request, whose report type results in a 207 (Multi-Status) [RFC4918] response, contains a Prefer header field with a preference of "return=minimal", the server SHOULD omit all DAV:propstat XML elements containing a DAV:status XML element of value 404 (Not Found) [I-D.ietf-httpbis-p2-semantics] from the 207 (Multi-Status) [RFC4918] response. If the omission of such a DAV:propstat element would result in a DAV:response XML element containing zero DAV:propstat elements, then the server MUST substitute a DAV:propstat element consisting of an empty DAV:prop element and a DAV:status element of value 200 (OK) [I-D.ietf-httpbis-p2-semantics] in its place.

2.2.1. Example: Typical REPORT request/response

This example uses the CALDAV:calendar-multiget [RFC4791] REPORT type.

>> Request <<

REPORT /murch/work/ HTTP/1.1

Host: caldav.example.com

Content-Type: application/xml; charset=utf-8

Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>

<C:calendar-multiget xmlns:C="urn:ietf:params:xml:ns:caldav"

xmlns:D="DAV:"

xmlns:X="http://ns.example.com/foobar/">

<D:prop>

<D:getetag/>

<X:foobar/>

</D:prop>

<D:href>/murch/work/abc.ics</D:href>

<D:href>/murch/work/qrs.ics</D:href>

<D:href>/murch/work/xyz.ics</D:href>

</C:calendar-multiget>

>> Response <<

HTTP/1.1 207 Multi-Status

Content-Type: application/xml; charset=utf-8

Content-Length: xxxx

Vary: Prefer

```
<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:C="urn:ietf:params:xml:ns:caldav"
  xmlns:D="DAV:"
  xmlns:X="http://ns.example.com/foobar/">
  <D:response>
    <D:href>/murch/work/abc.ics</D:href>
    <D:propstat>
      <D:prop>
        <D:getetag>"jahsd823ru"</D:getetag>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
    <D:propstat>
      <D:prop>
        <X:foobar/>
      </D:prop>
      <D:status>HTTP/1.1 404 Not Found</D:status>
    </D:propstat>
  </D:response>
  <D:response>
    <D:href>/murch/work/qrs.ics</D:href>
    <D:status>HTTP/1.1 404 Not Found</D:status>
  </D:response>
  <D:href>/murch/work/xyz.ics</D:href>
  <D:propstat>
    <D:prop>
      <D:getetag>"p08ulkj"</D:getetag>
    </D:prop>
    <D:status>HTTP/1.1 200 OK</D:status>
  </D:propstat>
  <D:propstat>
    <D:prop>
      <X:foobar/>
    </D:prop>
    <D:status>HTTP/1.1 404 Not Found</D:status>
  </D:propstat>
</D:multistatus>
```


2.2.2. Example: Minimal REPORT request/response

This example uses the CALDAV:calendar-multiget [[RFC4791](#)] REPORT type.

>> Request <<

```
REPORT /murch/work/ HTTP/1.1
```

```
Host: caldav.example.com
```

```
Content-Type: application/xml; charset=utf-8
```

```
Content-Length: xxxx
```

```
Prefer: return=minimal
```

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

```
<C:calendar-multiget xmlns:C="urn:ietf:params:xml:ns:caldav"
```

```
    xmlns:D="DAV:"
```

```
    xmlns:X="http://ns.example.com/foobar/">
```

```
  <D:prop>
```

```
    <D:getetag/>
```

```
    <X:foobar/>
```

```
  </D:prop>
```

```
  <D:href>/murch/work/abc.ics</D:href>
```

```
  <D:href>/murch/work/qrs.ics</D:href>
```

```
  <D:href>/murch/work/xyz.ics</D:href>
```

```
</C:calendar-multiget>
```


>> Response <<

```
HTTP/1.1 207 Multi-Status
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx
Preference-Applied: return=minimal
Vary: Prefer

<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:C="urn:ietf:params:xml:ns:caldav"
                xmlns:D="DAV:">
  <D:response>
    <D:href>/murch/work/abc.ics</D:href>
    <D:propstat>
      <D:prop>
        <D:getetag>"jahsd823ru"</D:getetag>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
  <D:response>
    <D:href>/murch/work/qrs.ics</D:href>
    <D:status>HTTP/1.1 404 Not Found</D:status>
  </D:response>
  <D:href>/murch/work/xyz.ics</D:href>
  <D:propstat>
    <D:prop>
      <D:getetag>"p08ulkj"</D:getetag>
    </D:prop>
    <D:status>HTTP/1.1 200 OK</D:status>
  </D:propstat>
</D:response>
</D:multistatus>
```

2.3. Minimal PROPPATCH Response

When a PROPPATCH [[RFC4918](#)] request contains a Prefer [[I-D.snell-http-prefer](#)] header field with a preference of "return=minimal", and all instructions are processed successfully, the server SHOULD return a 200 (OK) [[I-D.ietf-httpbis-p2-semantics](#)] response with an empty (zero-length) message body instead of a 207 (Multi-Status) [[RFC4918](#)] response.

2.3.1. Example: Typical PROPPATCH request/response

>> Request <<

```
PROPPATCH /container/ HTTP/1.1
Host: webdav.example.com
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8"?>
<D:propertyupdate xmlns:D="DAV:">
  <D:set>
    <D:prop>
      <D:displayname>My Container</D:displayname>
    </D:prop>
  </D:set>
</D:propertyupdate>
```

>> Response <<

```
HTTP/1.1 207 Multi-Status
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>/container/</D:href>
    <D:propstat>
      <D:prop>
        <D:displayname/>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```


2.3.2. Example: Minimal PROPPATCH request/response

>> Request <<

```
PROPPATCH /container/ HTTP/1.1
Host: webdav.example.com
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx
Prefer: return=minimal

<?xml version="1.0" encoding="utf-8"?>
<D:propertyupdate xmlns:D="DAV:">
  <D:set>
    <D:prop>
      <D:displayname>My Container</D:displayname>
    </D:prop>
  </D:set>
</D:propertyupdate>
```

>> Response <<

```
HTTP/1.1 200 OK
Content-Length: 0
Preference-Applied: return=minimal
```

2.4. Minimal MKCALENDAR / MKCOL Response

Both the MKCALENDAR [[RFC4791](#)] and Extended MKCOL [[RFC5689](#)] specifications indicate that a server MAY return a message body in response to a successful request. This specification explicitly defines the intended behavior in the presence of the Prefer [[I-D.snell-http-prefer](#)] header field.

When a MKCALENDAR or an Extended MKCOL request contains a Prefer header field with a preference of "return=minimal", and the collection is created with all requested properties being set successfully, the server SHOULD return a 201 (Created) [[I-D.ietf-httpbis-p2-semantics](#)] response with an empty (zero-length) message body.

[2.4.1.1.](#) Example: Verbose MKCOL request/response

>> Request <<

```
MKCOL /container/ HTTP/1.1
Host: webdav.example.com
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8"?>
<D:mkcol xmlns:D="DAV:">
  <D:set>
    <D:prop>
      <D:displayname>My Container</D:displayname>
    </D:prop>
  </D:set>
</D:mkcol>
```

>> Response <<

```
HTTP/1.1 201 Created
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8"?>
<D:mkcol-response xmlns:D="DAV:">
  <D:propstat>
    <D:prop>
      <D:displayname/>
    </D:prop>
    <D:status>HTTP/1.1 200 OK</D:status>
  </D:propstat>
</D:mkcol-response>
```


2.4.2. Example: Minimal MKCOL request/response

>> Request <<

```
MKCOL /container/ HTTP/1.1
Host: webdav.example.com
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx
Prefer: return=minimal

<?xml version="1.0" encoding="utf-8"?>
<D:mkcol xmlns:D="DAV:">
  <D:set>
    <D:prop>
      <D:displayname>My Container</D:displayname>
    </D:prop>
  </D:set>
</D:mkcol>
```

>> Response <<

```
HTTP/1.1 201 Created
Content-Length: 0
Preference-Applied: return=minimal
```

3. Reducing WebDAV Round-Trips with "return=representation"

The PUT, COPY, MOVE, [[RFC4918](#)] and POST [[RFC5689](#)] methods can be used to create or update a resource. In some instances, such as with CalDAV Scheduling [[RFC6638](#)], the created or updated resource representation may differ from the representation sent in the body of the request or referenced by the effective request URI. In cases where the client would normally issue a subsequent GET request to retrieve the current representation of the resource, the client SHOULD instead include a Prefer header field with the "return=representation" preference in the PUT, COPY, MOVE, or POST request. By doing this, the client can coalesce the create/update and retrieve operations into one round-trip rather than two.

3.1. Example: Typical resource creation and retrieval via POST + GET

>> Request <<

```
POST /murch/work;add-member/ HTTP/1.1
Host: caldav.example.com
Content-Type: text/calendar; charset=utf-8
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VEVENT
UID:CD87465FA
SEQUENCE:0
DTSTAMP:20120602T185254Z
DTSTART:20120602T160000Z
DTEND:20120602T170000Z
TRANSP:OPAQUE
SUMMARY:Lunch
ORGANIZER;CN="Ken Murchison":mailto:murch@example.com
ATTENDEE;CN="Ken Murchison";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
  mailto:murch@example.com
ATTENDEE;CN="John Doe";CUTYPE=INDIVIDUAL;PARTSTAT
  =NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:jdoe@
  example.com
END:VEVENT
END:VCALENDAR
```

>> Response <<

```
HTTP/1.1 201 Created
Location: /murch/work/abc.ics
Content-Length: 0
ETag: "nahduyejc"
Schedule-Tag: "jfd84hgbcn"
```

>> Request <<

```
GET /murch/work/abc.ics HTTP/1.1
Host: caldav.example.com
```


>> Response <<

```
HTTP/1.1 200 OK
Content-Type: text/calendar; charset=utf-8
Content-Length: xxxx
ETag: "nahduyejc"
Schedule-Tag: "jfd84hgbcn"

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Server//EN
BEGIN:VEVENT
UID:CD87465FA
SEQUENCE:0
DTSTAMP:20120602T185300Z
DTSTART:20120602T160000Z
DTEND:20120602T170000Z
TRANSP:OPAQUE
SUMMARY:Lunch
ORGANIZER;CN="Ken Murchison":mailto:murch@example.com
ATTENDEE;CN="Ken Murchison";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
  mailto:murch@example.com
ATTENDEE;CN="John Doe";CUTYPE=INDIVIDUAL;PARTSTAT
  =NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE;SCHEDULE-STATUS=
  1.2:mailto:jdoe@example.com
END:VEVENT
END:VCALENDAR
```


3.2. Example: Streamlined resource creation and retrieval via POST

>> Request <<

```
POST /murch/work;add-member/ HTTP/1.1
Host: caldav.example.com
Content-Type: text/calendar; charset=utf-8
Content-Length: xxxx
Prefer: return=representation

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VEVENT
UID:CD87465FA
SEQUENCE:0
DTSTAMP:20120602T185254Z
DTSTART:20120602T160000Z
DTEND:20120602T170000Z
TRANSP:OPAQUE
SUMMARY:Lunch
ORGANIZER;CN="Ken Murchison":mailto:murch@example.com
ATTENDEE;CN="Ken Murchison";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
  mailto:murch@example.com
ATTENDEE;CN="John Doe";CUTYPE=INDIVIDUAL;PARTSTAT
  =NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:jdoe@
  example.com
END:VEVENT
END:VCALENDAR
```


>> Response <<

```
HTTP/1.1 201 Created
Location: /murch/work/abc.ics
Content-Type: text/calendar; charset=utf-8
Content-Length: xxxx
Location: /murch/work/abc.ics
ETag: "nahduyejc"
Schedule-Tag: "jfd84hgbcn"
Preference-Applied: return=representation

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Server//EN
BEGIN:VEVENT
UID:CD87465FA
SEQUENCE:0
DTSTAMP:20120602T185300Z
DTSTART:20120602T160000Z
DTEND:20120602T170000Z
TRANSP:OPAQUE
SUMMARY:Lunch
ORGANIZER;CN="Ken Murchison":mailto:murch@example.com
ATTENDEE;CN="Ken Murchison";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
  mailto:murch@example.com
ATTENDEE;CN="John Doe";CUTYPE=INDIVIDUAL;PARTSTAT
  =NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE;SCHEDULE-STATUS=
  1.2:mailto:jdoe@example.com
END:VEVENT
END:VCALENDAR
```

4. The "depth-noroot" Processing Preference

The "depth-noroot" preference indicates that the client wishes for the server to exclude the target (root) resource from processing by the WebDAV method and only apply the WebDAV method to the target resource's subordinate resources.

depth-noroot = "depth-noroot"

This preference is only intended to be used with WebDAV methods whose definitions explicitly provide support for the Depth [[RFC4918](#)] header field. Furthermore, this preference only applies when the Depth header field has a value of "1" or "infinity" (either implicitly or explicitly).

The "depth-noroot" preference MAY be used in conjunction with the "return=minimal" preference in a single request.

[4.1.](#) Example: Typical PROPFIND request/response with Depth:1

This example fetches the DAV:sync-token [[RFC6578](#)] property for a collection and its child collections.

>> Request <<

```
PROPFIND /murch/ HTTP/1.1
Host: dav.example.com
Content-Type: text/xml
Content-Length: xxx
Depth: 1
```

```
<?xml version="1.0" encoding="UTF-8"?>
<D:propfind xmlns:D="DAV:">
  <D:prop>
    <D:sync-token/>
  </D:prop>
</D:propfind>
```


>> Response <<

HTTP/1.1 207 Multi-Status

Content-Type: application/xml; charset=utf-8

Content-Length: xxxx

```
<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>/murch/</D:href>
    <D:propstat>
      <D:prop>
        <D:sync-token>http://example.com/ns/sync/2216-2</D:sync-token>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
  <D:response>
    <D:href>/murch/work/</D:href>
    <D:propstat>
      <D:prop>
        <D:sync-token>http://example.com/ns/sync/2136-34</D:sync-token>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
  <D:response>
    <D:href>/murch/home/</D:href>
    <D:propstat>
      <D:prop>
        <D:sync-token>http://example.com/ns/sync/2141-19</D:sync-token>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```

4.2. Example: PROPFIND request/response with Depth:1 and Prefer:depth-noroot

This example fetches the DAV:sync-token [[RFC6578](#)] property for just the child collections.

>> Request <<

```
PROPFIND /murch/ HTTP/1.1
Host: dav.example.com
Content-Type: text/xml
Content-Length: xxx
Depth: 1
Prefer: depth-noroot
```

```
<?xml version="1.0" encoding="UTF-8"?>
<D:propfind xmlns:D="DAV:">
  <D:prop>
    <D:sync-token/>
  </D:prop>
</D:propfind>
```

>> Response <<

```
HTTP/1.1 207 Multi-Status
Content-Type: application/xml; charset=utf-8
Content-Length: xxxx
Preference-Applied: depth-noroot
Vary: Prefer
```

```
<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>/murch/work</D:href>
    <D:propstat>
      <D:prop>
        <D:sync-token>http://example.com/ns/sync/2136-34</D:sync-token>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
  <D:response>
    <D:href>/murch/home</D:href>
    <D:propstat>
      <D:prop>
        <D:sync-token>http://example.com/ns/sync/2141-19</D:sync-token>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```


5. Security Considerations

No new security considerations are introduced by use of the Prefer header field with WebDAV request methods, beyond those discussed in [[I-D.snell-http-prefer](#)] and those already inherent in those methods.

6. IANA Considerations

The following preference is to be added to the Preferences Registry defined in [[I-D.snell-http-prefer](#)].

- o Preference: depth-noroot
- o Description: The "depth-noroot" preference indicates that the client wishes for the server to exclude the target (root) resource from processing by the WebDAV method and only apply the WebDAV method to the target resource's subordinate resources.
- o Reference: [Section 4](#)
- o Notes: This preference is only intended to be used with WebDAV methods whose definitions explicitly provide support for the "Depth" [[RFC4918](#)] header field. Furthermore, this preference only applies when the "Depth" header field has a value of "1" or "infinity" (either implicitly or explicitly).

7. Acknowledgements

The author would like to thank the following individuals for contributing their ideas and support for writing this specification: Cyrus Daboo, Helge Hess, Andrew McMillan, and Arnaud Quillaud.

The author would also like to thank the Calendaring and Scheduling Consortium for advice with this specification, and for organizing interoperability testing events to help refine it.

8. References

8.1. Normative References

[I-D.ietf-httpbis-p2-semantic]
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content",
[draft-ietf-httpbis-p2-semantic-21](#) (work in progress),
October 2012.

[I-D.snell-http-prefer]
Snell, J., "Prefer Header for HTTP",

[draft-snell-http-prefer-18](#) (work in progress),
January 2013.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [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.
- [RFC4918] Dusseault, L., "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", [RFC 4918](#), June 2007.
- [RFC5689] Daboo, C., "Extended MKCOL for Web Distributed Authoring and Versioning (WebDAV)", [RFC 5689](#), September 2009.
- [RFC5995] Reschke, J., "Using POST to Add Members to Web Distributed Authoring and Versioning (WebDAV) Collections", [RFC 5995](#), September 2010.

[8.2.](#) Informative References

- [MSDN.aa493854]
Microsoft Developer Network, "PROPPATCH Method",
June 2006.
- [MSDN.aa563501]
Microsoft Developer Network, "Brief Header", June 2006.
- [MSDN.aa563950]
Microsoft Developer Network, "Depth Header", June 2006.
- [MSDN.aa580336]
Microsoft Developer Network, "PROPFIND Method", June 2006.
- [RFC4791] Daboo, C., Desruisseaux, B., and L. Dusseault,
"Calendaring Extensions to WebDAV (CalDAV)", [RFC 4791](#),
March 2007.
- [RFC6352] Daboo, C., "CardDAV: vCard Extensions to Web Distributed
Authoring and Versioning (WebDAV)", [RFC 6352](#), August 2011.
- [RFC6578] Daboo, C. and A. Quillaud, "Collection Synchronization for
Web Distributed Authoring and Versioning (WebDAV)",
[RFC 6578](#), March 2012.
- [RFC6638] Daboo, C. and B. Desruisseaux, "Scheduling Extensions to

CalDAV", [RFC 6638](#), June 2012.

Appendix A. The Brief and Extended Depth Request Header Fields

This document is based heavily on the Brief [[MSDN.aa563501](#)] and extended Depth [[MSDN.aa563950](#)] request header fields. The behaviors described in [Section 2.1](#) and [Section 2.3](#) are identical to those provided by the Brief header field when used with the PROPFIND [[MSDN.aa580336](#)] and PROPPATCH [[MSDN.aa493854](#)] methods respectively. The behavior described in [Section 4](#) is identical to that provided by the "1,noroot" [[MSDN.aa563950](#)] and "infinity,noroot" [[MSDN.aa563950](#)] Depth header field values.

Authors are encouraged to implement the Brief header field functionality in conjunction with this specification to further promote interoperability with products that use the Brief header field exclusively.

Appendix B. Open Issues

- o Is the Vary header field necessary in the PROPFIND/REPORT responses? Are PROPFIND/REPORT results ever cached?

Appendix C. Change Log (to be removed by RFC Editor before publication)

C.1. Since -00

- o Updated to comply with [draft-snell-httpprefer-18](#).
- o Reordered "Minimal REPORT Response" and "Minimal PROPPATH Response" sections.
- o Added some explanatory text to examples.

C.2. Since CalConnect XXIV

- o Updated references.
- o Stated that "depth-noroot" can be used in conjunction with "return=minimal".
- o Added text mentioning that "depth-noroot" is based on the MSDN "1,noroot" and "infinity,noroot" Depth header values.
- o The server behavior required when "return=minimal" would result in zero DAV:propstat elements has been changed

from:

```
<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>/container/</D:href>
    <D:status>HTTP/1.1 200 OK</D:status>
  </D:response>
</D:multistatus>
```

to the slightly more verbose:

```
<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>/container/</D:href>
    <D:propstat>
      <D:prop/>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```

Author's Address

Kenneth Murchison
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
US

Email: murch@andrew.cmu.edu

