

WebDAV
Internet Draft
Document: [draft-ietf-webdav-properties-extension-00.txt](#) September 1999
Category: Informational
Expires: March, 2000

J. Amsden

IBM

Proposed Extensions to WebDAV Properties

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#) [1].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. 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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

1. Abstract

The WebDAV protocol defines facilities for adding properties to Web resources. These properties consist of name/value pairs interchanged using XML. The protocol includes the PROPFIND method for accessing properties, and the PROPPATCH method for updating and removing properties. Experience building both WebDAV clients and a WebDAV server indicates the need for three minor extensions to the current properties protocol: 1) allow PROPPATCH to create and initialize the properties of a resource that did not exist, 2) distinguish between adding a new property, and setting the value of an existing property of a resource, and 3) give client applications more control in specifying how PROPPATCH errors should be handled.

2. Conventions used in this document

Since this document describes a set of extensions to the HTTP/1.1 protocol, the augmented BNF used here to describe protocol an element is exactly the same as described in Section 2.1 of [[HTTP](#)]. Since

this augmented BNF uses the basic production rules provided in Section 2.2 of [[HTTP](#)], these rules apply to this document as well.

Amsden

Informational ð Expiration March 2000

1

Proposed Extensions to WebDAV Properties September 1999

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

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 [RFC-2119](#) [2].

[3. Terminology](#)

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

Null Property - A property that responds with 404 (Not Found) to a PROPFIND method. A null property MUST not appear in the list of properties of a resource the PROPFIND is specified with DAV:allprop.

[4. Introduction](#)

This Internet Draft specification introduces three simple extensions or modifications to the handling of WebDAV properties. All are based on experience gained through building a WebDAV class 2 server, and sample client applications. The extensions are all compatible with existing WebDAV property semantics, and all involve property updates through the PROPPATCH method.

[4.1 PROPPATCH on a Null Resource](#)

HTTP and WebDAV do not specify methods for creating resources; they are created as the side effect of other methods. For example, PUT on a null resource (a resource that does not exist) creates the resource and establishes its initial contents. LOCK on a null resource creates a lock-null resource, which has properties, but no content. COPY and MOVE can create resources at the destination URI. It is often necessary to create resources and initialize their state before any other access is allowed in order to ensure the integrity of the operations. WebDAV currently has no protocol for creating and

initializing the properties of a resource in a single method. This Internet Draft proposes that PROPPATCH on a null resource creates the resource with empty contents, and initializes its properties based on the propertyupdate elements in the request entity body.

The WebDAV spec is currently somewhat silent about PROPPATCH on a null resource. It does say that PROPPATCH on a lock null resource (a resource created by the LOCK method) will fail, but this is inconsistent with PUT which is allowed, and changes the state of the resource from lock null to resource. [[HTTP](#)] uses status 404, Not Found, to indicate the server has not found anything matching the Request-URI. It does not say that the server cannot create a resource

<Lastname>

Category - Expiration

2

Proposed Extensions to WebDAV Properties September 1999

at the Request-URI. The HTTP and WebDAV PUT method on a null resource is defined to create the resource and set its initial contents. Allowing PROPPATCH on a null resource to create the resource and set its initial properties is consistent with the PUT semantics, and other resource state changing methods that create resources as a side effect.

[4.2](#) Adding New Properties

Typical database management systems distinguish three kinds of updates; add, remove, and change. It is often necessary to distinguish between add and change in order to ensure a property is added only once, and that its value is not inadvertently changed after it has been initialized.

The WebDAV PROPPATCH method updates a resource's properties based on the instructions contained in the DAV:propertyupdate element of the request entity body. The propertyupdate element currently supports setting and removing properties. DAV:set creates a new property if it does not exist, and sets its value. If the property already exists, the value is changed.

This Internet Draft proposes the addition of a DAV:add propertyupdate that adds a new property and sets its value, but fails if the property already exists. In addition, this draft proposes that DAV:set on a null property (a property that does not exist) fails instead of creating a new property. This may help prevent erroneous client property updates resulting from setting the values of properties where the property name is incorrectly specified.

4.3 Handling PROPPATCH Errors

Client applications currently use PROPPATCH to update the properties of a resource. PROPPATCH supports the update of a number of properties in a single method request in order to reduce communication overhead and maintain resource state integrity with atomic updates.

Currently, the WebDAV spec dictates that the result of a PROPPATCH is an atomic transaction. That is, either all the updates succeed, or none of them succeed and the updates are rolled back. This can result in excessive client burden. A client application may attempt to make a number of updates with PROPPATCH, and doesn't care if some of them don't succeed. This is often the case when some of the properties are live on the server, but the client has no way to determine which are live without attempting to update their value and getting an error. This situation arises when different servers support different live properties. Updating the desired properties, ignoring the ones that are live requires the client to attempt the update, examine the DAV:multistatus response that is returned, construct a new PROPPATCH entity request body removing the properties that failed, and re-submitting the PROPPATCH request.

<Lastname>	Category - Expiration	3
------------	-----------------------	---

Proposed Extensions to WebDAV Properties September 1999

This Internet Draft proposes that a mechanism be added to the DAV:propertyupdate element that gives client applications more control over updating resource properties by allowing the protocol to specify how to handle errors on each of the updates in the DAV:propertyupdate of a PROPPATCH entity request body. This is similar to the DAV:propertybehavior element of the COPY and MOVE entity request body that is used to specify how properties are copied to the destination resource.

By making this change to the PROPPATCH entity request body, it is now possible to completely specify the semantics of COPY and MOVE in terms of the more primitive GET, PUT, PROPFIND, and PROPPATCH methods. This is not currently possible because there is no way to process the DAV:propertybehavior element of the COPY or MOVE entity request body with a PROPPATCH method.

5. PROPPATCH Protocol Changes

This section specifies changes to [section 8.2](#) PROPPATCH of [[WebDAV](#)] and [section 12](#) to support the semantics described above.

Update [section 8.2](#) PROPPATCH to indicate that PROPPATCH on a null resource creates the resource at the given Request-URI, and initializes its properties as specified in the DAV:propertyupdate element of the PROPPATCH entity request body.

Change [section 8.2](#) PROPPATCH to include simple error processing of PROPPATCH method requests. Update the DAV:set and DAV:remove elements to include information describing how the client wishes to handle errors. Change the semantics of DAV:set to fail on a null property.

Change [section 12.13.2](#) set XML element to:

12.13.2 set XML element

Name: set
Namespace: DAV:
Purpose: List the DAV property values to be set for a resource.

Description: The set XML element MUST contain only a prop XML element. The elements contained by the prop XML element inside the set XML element MUST specify the name and value of properties that are set on the resource identified by the Request-URI. If the property does not exist then a 404 (Not Found) status is returned, and the property update fails. The DAV:updatebehavior element specifies how update failures are to be handled. Language tagging information in the property's value (in the "xml:lang" attribute, if present) MUST be persistently stored along with the property, and MUST be subsequently retrievable using PROPFIND.

<!ELEMENT set (prop, updatebehavior?) >

<Lastname> Category - Expiration 4

Proposed Extensions to WebDAV Properties September 1999

Change [section 12.13.1](#) remove XML element to:

12.13.1 remove XML element

Name: remove
Namespace: DAV:
Purpose: List the DAV property values to be removed from a resource.

Description: Remove instructs that the properties specified in the prop should be removed. Specifying the removal of a null property is not an error. All the XML elements in a prop XML element inside of a

remove XML element MUST be empty, as only the names of properties to be removed are required. The DAV:updatebehavior element specifies how update failures are to be handled.

<!ELEMENT remove (prop, updatebehavior?) >

Extend the DAV:propertyupdate element of the PROPPATCH entity request body to include element DAV:add. The DAV:add element has the same contents as DAV:set.

Adding a property fails with 403 (Forbidden) if the property already exists. Setting a null property results in a 404 (Not Found) status.

Change to [section 12.13](#) propertyupdate XML element to:

12.13 propertyupdate XML element

Name: propertyupdate
Namespace: DAV:
Purpose: Contains a request to alter the properties on a resource
Description: This XML element is a container for the information required to modify the properties on the resource. This XML element is multi-valued to support adding, changing, and/or removing properties.

<!ELEMENT propertyupdate (add | set | remove) >

Add [section 12.13.3](#), add XML element as follows:

12.13.3 add XML element

Name: add
Namespace: DAV:
Purpose: List the DAV property values to be added to a resource.
Description: The add XML element MUST contain only a prop XML element. The elements contained by the prop XML element inside the add XML element MUST specify the name and value of properties that are to be added to the resource identified by the Request-URI. If a property already exists then a 403 (Forbidden) status is returned, and the property is not added. The DAV:updatebehavior element

<Lastname> Category - Expiration 5

Proposed Extensions to WebDAV Properties September 1999

specifies how update failures are to be handled. Language tagging information in the property's value (in the "xml:lang" attribute, if present) MUST be persistently stored along with the property, and

MUST be subsequently retrievable using PROPFIND.

<!ELEMENT add (prop, propertyupdate?) >

Add sections to [section 12.13](#) to specify error processing on property updates.

12.13.4 updatebehavior XML element

Name: updatebehavior

Namespace: DAV:

Purpose: Specify how errors on property updates are to be handled.

Description: An updatebehavior specifies how errors are to be handled on the corresponding DAV:add, DAV:set, or DAV:remove propertyupdate. The client can specify that the error should be ignored, or that it must succeed. If DAV:mustsucceed is specified, then the PROPPATCH method will fail if any of the updates fail. This is the default behavior.

<!ELEMENT updatebehavior (ignore | mustsucceed) >

12.13.5 ignore XML element

Name: ignore

Namespace: DAV:

Purpose: Causes corresponding property update errors to be ignored

Description: The default behavior for a PROPPATCH method is all property updates must succeed, or none of them succeed. If an updatebehavior is not included, it is equivalent to the default behavior or specifying <mustsucceed>*</mustsucceed> meaning that all the updates must be successful or none of them are performed. The DAV:ignore element specifies that the server should make best-effort property updates. Any error caused by the associated propertyupdate is ignored. The error is reported in the resulting DAV:multistatus, but the rest of the updates specified in the PROPPATCH entity request body are processed as if this propertyupdate was not specified.

<!ELEMENT ignore EMPTY >

12.13.6 mustsucceed XML element

Name: mustsucceed

Namespace: DAV:

Purpose: Specifies the corresponding property update must succeed.

Description: The default behavior for a PROPPATCH method is all property updates must succeed, or none of them succeed. The DAV:mustsucceed element allows the client to specify a list of named

properties whose property update must succeed. Any error caused by the associated propertyupdate causes the PROPPATCH method to fail, and all property updates are rolled back. If a value of "*" is given for the mustsucceed XML element, this designates that all property updates must succeed. "*" is the only PCDATA value that can be specified.

```
<!ELEMENT mustsucceed (#PCDATA | href+)>
```

Note: element DAV:mustsucceed may not be necessary because the default behavior is for the PROPPATCH to fail if any propertyupdate fails. It is really only necessary to specify which errors are to be ignored. I have included it here for completeness, and consistency with the DAV:keepalive XML element.

6. Formal Syntax

< Commonly used grammar is BNF grammar defined in [RFC-2234](#). Suggested wording.>

This section summarizes the changes and additions to the WebDAV DTD given in section 24.1 of [[WebDAV](#)].

```
<!ELEMENT propertyupdate (add | set | remove)+>
```

```
<!ELEMENT add (prop, updatebehavior?) >
```

```
<!ELEMENT set (prop, updatebehavior?) >
```

```
<!ELEMENT remove (prop, updatebehavior?) >
```

```
<!ELEMENT updatebehavior (ignore | mustsucceed)>
```

```
<!ELEMENT ignore EMPTY>
```

```
<!ELEMENT mustsucceed (#PCDATA | href+)>
```

7. Security Considerations

This section is provided to detail issues concerning security implications of which WebDAV applications need to be aware.

All of the security considerations of HTTP/1.1 and the WebDAV Distributed Authoring Protocol specification also apply to WebDAV collections.

These changes may introduce a denial of service security violation through the possibility of a client application exhausting server resources with PROPPATCH methods. However, the same possibility

exists with PUT operations, so no new security issues are introduced.

8. IANA Considerations

<Lastname>	Category - Expiration	7
------------	-----------------------	---

Proposed Extensions to WebDAV Properties September 1999

This document uses the namespaces defined by [[WebDAV](#)] for properties and XML elements. All other IANA considerations mentioned in [[WebDAV](#)] also apply to this document.

9. Copyright

To be supplied by the RFC Editor.

10. Intellectual Property

To be supplied by the RFC Editor.

11. References

- 1 Bradner, S., "The Internet Standards Process -- Revision 3", [BCP 9](#), [RFC 2026](#), October 1996.
- 2 Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997

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

[XML] T. Bray, J. Paoli, C.M. Sperberg-McQueen, "Extensible Markup Language (XML)." World Wide Web Consortium Recommendation REC-xml-19980210. <http://www.w3.org/TR/1998/REC-xml-19980210>.

[HTTP] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1." [RFC 2068](#). UC Irvine, DEC,

MIT/LCS. January, 1997.

[WebDAV] Y. Y. Goland, E. J. Whitehead, Jr., A. Faizi, S. R. Carter,
D.

Jensen, "HTTP Extensions for Distributed Authoring - WebDAV." [RFC
2518](#).

Microsoft, U.C. Irvine, Netscape, Novell. February, 1999.

11. Acknowledgments

This draft has benefited from thoughtful discussion by Jim Whitehead,
Geoff Clemm, and others.

12. Author's Addresses

James Amsden
IBM

<Lastname>	Category - Expiration	8
------------	-----------------------	---

Proposed Extensions to WebDAV Properties September 1999

3039 Cornwallis Road
RPT, NC
Phone: 919-461-3919
Email: jamsden@us.ibm.com

<Lastname>

Category - Expiration

9