

Additional WebDAV Collection Properties

Status of this Memo

This document is an Internet-Draft. 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."

To view the entire list of current Internet-Drafts, please check the "l1d-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern Europe), ftp.nic.it (Southern Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

A revised version of this draft document may be submitted to the RFC editor as a Proposed Standard for the Internet Community. Discussion and suggestions for improvement are requested. This document will expire before June, 1999. Distribution of this draft is unlimited.

Abstract

The WebDAV protocol defines a basic set of properties that should be understood by all DAV servers. This document defines a small set of additional properties that are very useful for displaying navigation interfaces for WebDAV enabled web sites.

Table of Contents

1.	Introduction	2
1.1.	Notational Conventions	2
2.	The Collection Properties	2
1.2.	The childcount Property	2
1.3.	The defaultdocument Property	3
1.4.	The id Property	3
1.5.	The iscollection Property	3
1.6.	The ishidden Property	4
1.7.	The isstructureddocument Property	4
1.8.	The hassubs Property	5
1.9.	The nosubs Property	5

1.10.	The objectcount property	6
1.11.	The reserved Property	6
1.12.	The visiblecount Property	6
3.	Security Considerations	6
4.	References	7
5.	Author's Addresses	7

[1.](#) Introduction

Collections are a type of resource supported by DAV that may have special properties. The DAV draft [[DAV](#)] defines how to interact with collections using methods such as MKCOL and PROPFIND, and defines a number of properties which were essential to the proper use of collections.

The properties in this draft are not as essential to using collections, therefore they are optional. However, the authors find these to be useful properties to clients, especially to save time and processing and to know how to display collections.

Many of these properties are merely hints to a client user interface to help it display collections and options on collections in a manner that is most helpful to the user.

[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 [RFC 2119](#) [[KEYWORDS](#)].

[2.](#) The Collection Properties

This document defines an additional set of properties for use by the WebDAV [1] protocol. These properties provide hints that make it easier to provide efficient user interfaces that provide navigation for WebDAV enabled web sites.

If this set of properties is supported, then ALL the properties listed in this draft MUST be supported.

This document follows the same collections as the WebDAV protocol specification [1] for defining these properties.

[1.2.](#) The childcount Property

Name:	childcount
Namespace:	DAV:
Purpose:	Specifies the number of contained resources.
Description:	This property identifies the number of resources contained in a given collection. It contains a single

integer value with the count of contained resources.
This property includes child collections in the count.

Definition: <![ELEMENT childcount (#PCDATA) >

1.3. The defaultdocument Property

Name: defaultdocument
Namespace: DAV:
Purpose: Specifies the default document for a collection.
Description: This property contains a URL that identifies the default document for a collection. This is intended for collection owners to be able to set a default document, for example index.html or default.html. If this property is absent, other means must be found to determine the default document.
If this property is present but null, the collection does not have a default document and the collection member listing should be used (or nothing).
Notes: The server implementation does not need to store this property in the normal property store (the property could well be live).

<![ELEMENT defaultdocument (#PCDATA) >

1.4. The id Property

Name: id
Namespace: DAV:
Purpose: Specifies a unique identifier for this resource.
Description: This property contains a globally unique string that identifies this resource. This property MUST be unique across the entire Internet. The id property does not change if the resource changes. This property is intended to aid in recognition of a resource even when moved, updated or renamed. The value of this property is a URI. The URI could contain a string UUID as defined by ISO for RPC [[RPC](#)] but any unique URI is sufficient.
Definition: <![ELEMENT id (#PCDATA) >

1.5. The isfolder Property

Name: isfolder
Namespace: DAV:
Purpose: Specifies whether or not a collection should appear as a folder.
Description: This property identifies whether or not a collection should appear as a folder. It contains either the values "1" or "0". If "1" or absent, the collection should be displayed as a folder. If "0", the collection should NOT be displayed as a folder. For example, a structured

document (see below) should have "isfolder" set to "0".
Definition: <!ELEMENT ishidden (#PCDATA) >

1.6. The ishidden Property

Name: ishidden
Namespace: DAV:
Purpose: Specifies whether or not a resource is hidden.
Description: This property identifies whether or not a resource is hidden. It contains either the values "1" or "0". This can be considered a hint to the client UI: under normal conditions, for non-expert users, hidden files should not be exposed to users. The server may omit the hidden resource from some presentational listings, otherwise the client is responsible for removing hidden resources when displaying to the user. If this property is absent, the collection is not hidden. Since this property provides no actual form of protection to the resources, this MUST NOT be used as a form of access control and should only be used for presentation purposes.
Usage example: Many file systems have the option to hide files from the user, but the user can, with special commands, override the hiding.
Definition: <!ELEMENT ishidden (#PCDATA) >

1.7. The isstructureddocument Property

Name: isstructureddocument
Namespace: DAV:
Purpose: Specifies whether the resource is a structured document.
Description: A structured document is a collection (iscollection should also be true), so COPY, MOVE and DELETE work as for a collection. The structured document may behave at times like a document. For example, clients may wish to display the resource as a document rather than as a collection. This contains either "1" (true) or "0". If this property is absent, the collection is not a structured document.
This property can also be considered a hint for the client UI: if the value of "isstructureddocument" is "1", then the client UI may display this to the user as if it were single document. This can be very useful when the default document of a collection is an HTML page with a bunch of images which are the other resources in the collection: only the default document is intended to be viewed as a document, so the entire structure can appear as one document.
A Structured document may contain collections. A structured document must have a default document (if the "defaultdocument" property is absent, the default

document is assumed by the client to be index.html).
Definition: `<!ELEMENT isstructureddocument (#PCDATA) >`

1.8. The hassubs Property

Name: hassubs
Namespace: DAV:
Purpose: Identifies whether this collection contains any collections which are folders (see "isfolder").
Description: This property identifies whether or not a folder contains sub-folders, from the point of view of client display. Sub-folders are child collections for which "isfolder" = "1".
The "hassubs" property contains either the values "1" (yes) or "0" (zero=no). If absent, nothing can be guessed about whether the collection has sub-folders. This property is useful for the efficient display of hierarchy user interfaces.
If "hassubs" = "1", then "isfolder" should also be "1" so that clients understand that the folder can be expanded to view its children.
Definition: `<!ELEMENT hassubs (#PCDATA) >`

1.9. The nosubs Property

Name: nosubs
Namespace: DAV:
Purpose: Identifies whether this collection allows child collections to be created.
Description: This property identifies whether or not a collection allows child collections to be created. It contains either the values "1" or "0" ("1" indicates that the collection does not allow child collections). While this data is redundant with that returned by the OPTIONS method, providing this information as a property allows better performance since the client can verify the behavior ahead of time without having to issue an individual OPTIONS request on every collection it encounters. If absent, nothing can be guessed about whether the collection allows sub-collections.
This property can also be considered to be a hint to the UI about displaying options to the user (the UI might eliminate the option to create a child collection). It is different from a "create child" access right, because the client UI may want to display a "create child collection" option without trying to find out if the user has permissions. This property can be used to suggest that creating child collections just doesn't make sense on this collection no matter what rights the user has. It is most useful on special-purpose collections, such as a deleted files collection or a

collection which represents a device such as a printer. This property should not be construed as meaning that sub-collections do not already exist on the collection - it simply prevents new collections from being created by the client.

Definition: `<!ELEMENT nosubs (#PCDATA) >`

1.10. The objectcount property

Name: objectcount

Namespace: DAV:

Purpose: To count the number of non-folder resources in the collection.

Description: This is different from childcount in that it omits counting child collections for which "isfolder" = "1".

Definition: `<!ELEMENT objectcount (#PCDATA) >`

1.11. The reserved Property

Name: reserved

Namespace: DAV:

Purpose: Specifies whether or not the collection is reserved.

Description: A reserved collection is one that is specially managed by the server and cannot be deleted, renamed, or moved by the client. It contains either the values "1" or "0". Attempts to MOVE or DELETE a reserved collection will fail, and this SHOULD be reflected in the client UI. If absent, the collection should NOT be reserved. The server may allow clients to set this property. It may make sense to also specify that this collection is reserved in the resourcetype; however, in most ways this behaves like a normal collection.

Definition: `<!ELEMENT reserved (#PCDATA) >`

1.12. The visiblecount Property

Name: visiblecount

Namespace: DAV:

Purpose: Counts the number of visible non-folder resources in the collection.

Description: This is the most immediately useful property for the client UI to use to display the sizes of collections for users. The client UI could also display progress when downloading a long list of children in a collection if it knows the total number in advance. This counts all children for which "ishidden" = "0" and "isfolder" = "0".

Definition: `<!ELEMENT reserved (#PCDATA) >`

3. Security Considerations

The ability to hide resources should not be considered a security feature. With the current described behavior, there is a danger that users will be able to discover the hidden resource by trying to write a file of the same name as the hidden resource. In addition, since hidden resources will be exposed via PROPFIND, there is no actual security for them, the hidden nature MUST simply be for presentation (user interface) purposes. This is because the hidden resource feature described here is not intended to be a security feature but a client display feature.

Solving the problem of having resources which are undetectable under certain conditions is beyond the scope of this draft.

4. References

[HTTP] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2068](#), U.C. Irvine, DEC, MIT/LCS, January 1997.

[DAV] Y. Goland, E.J. Whitehead, A. Faizi, S.R. Carter, D. Jenson, "Extensions for Distributed Authoring on the World Wide Web", April. 1998, internet-draft, work-in-progress, [draft-ietf-webdav-protocol-08](#).

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

[RPC] ISO/IEC 11578:1996(E) International Standard: "Information Technology -- Open Systems Interconnection -- Remote Procedure Call", December, 1996 (see Annex A)

5. Author's Addresses

Alex Hopmann
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
Email: alexhop@microsoft.com

Lisa Lippert
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
Email: lisal@microsoft.com