WEBDAV Working Group
INTERNET DRAFT
<draft-ietf-webdav-collection-reqts-05.txt>
Expires December 18, 1999

J. Slein, Xerox
J. Davis, CourseNet

June 18, 1999

Requirements for Advanced Collection Functionality in WebDAV

Status of this Memo

Abstract

The WebDAV Distributed Authoring Protocol [WebDAV] provides basic support for collections, offering the ability to create and list unordered collections. Many applications, however, need more powerful collections, especially for resource sharing and collection ordering.

This draft sets out requirements for more advanced, optional collection functionality. It extends the base functionality in the two directions of resource sharing and collection ordering. A separate WebDAV specification is expected to define protocol elements providing the functionality described here.

1 Terminology

The terminology used here follows and extends that 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].

Association
: A direct or indirect connection between a resource and a namespace element to support resource sharing. Bindings, URI mappings, and redirect references are types of associations.

URI Mapping
: An association between an absolute URL or URI and a resource. Since a resource can represent items that are not network retrievable, as well as those that are, it is possible for a resource to have zero, one, or many URI mappings to URLs or URIs. Mapping a resource to an "http" scheme URL makes it possible to submit HTTP protocol requests to the resource using the URL.

Path Segment
: Informally, the characters found between slashes ("/") in a URL or URI. Formally, as defined in section 3.3 of [URI].

Binding
: An association between a single path segment (in a collection) and a resource. A binding creates one or more URI mappings, and hence is a mechanism for resource sharing, allowing a single resource to be accessed from multiple locations in a URI namespace.

Collection
: A resource that contains, as part of its state, a set of bindings which identify member resources.

Internal Member URI
: The URI mapping, created by a binding, that is a member of a collection. While, in general, bindings can create multiple URI mappings to a resource, for a given request, only one of these URI mappings is referred to as the internal member. The URI of the parent collection used in a given request determines the base URI for internal member URI calculation.

2 Introduction and Rationale

The simple collections that the WebDAV Distributed Authoring Protocol specification supports are powerful enough to be widely useful. They provide for the hierarchical organization of resources, with mechanisms for creating and deleting collections,

copying and moving them, locking them, adding resources to them and deleting resources from them, and getting listings of their members. Delete, copy, move, list, and lock operations can be applied recursively, so that a client can operate on whole hierarchies with a single request.

Many applications, however, need more powerful collections. There are two areas in particular where more powerful functionality is often needed: shared resources and ordering. This draft details the additional functionality that is needed in these two areas.

In both areas, it should be a goal of the protocol specification to be compatible with HTTP 1.1 and with the WebDAV Distributed Authoring Protocol. It should be a goal that down-level clients be able to take advantage of any new constructs, even if they cannot create and manipulate them. It should be a goal that the

new functionality be relatively simple to implement, both for clients and for servers. It should be a goal that the new constructs satisfy the infrastructure needs of other WebDAV facilities, particularly the current work on versioning and configuration management.

2.1 Shared Resources

Associations make it possible for many collections, on the same or different servers, to share the same resource. Only one physical copy of the resource need exist, and any changes made in the resource are visible from all the collections that share it.

A number of scenarios motivate adding associations to the functionality of WebDAV:

Organizing resources into hierarchies places them into collections, which are more easily browsed and manipulated than a flat namespace. 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.

Sharing between collections on different servers may be desired. For example, the mathematics department at one university may wish to create a collection of information on fractals that contains some local resources, but also provides access to resources at several other universities. For many reasons, it may be

undesirable to make physical copies of the shared resources on the local server - to conserve disk space, to respect copyright constraints, or to make any changes in the shared resources visible automatically.

In another scenario, a manufacturing company develops and maintains its product maintenance manuals on the Web. There is a separate collection for each product manual. Each manual is divided into sections, one section for every product component. Since many of the companyÆs products contain some of the same components, many of the product maintenance manuals have sections in common. Each manual may have some unique sections, but for product components that are common to multiple products, the manual's collection accesses a resource in a shared library.

These requirements do not address issues of the integrity of associations, though integrity will be of great importance to some applications. Some applications cannot tolerate broken links. A software development application, for example, must be able to rely on the integrity of references to component modules. Other applications may want integrity not to be enforced. For example, a Web site manager might want to be able to create access paths before the resources are created for which they will provide access, or might want to be able to remove content temporarily

without deleting and later recreating the associations to the content. Addressing these concerns is considered too difficult for the planned protocol specification, however. Consequently, they are not included in this set of requirements.

2.2 Ordered Collections

For many applications, it is useful for clients to be able to impose orderings on collections at the server. When the server receives a request for a list of a collection's members, it always responds with a list ordered according to the ordering specified for that collection. In the product manual application above, the sections of each manual may be ordered so that they can be printed together as a book. A configuration management application might use a collection to represent a version series, in which case the "derives from" relationship might be represented as an ordering on the collection.

A collection ordering may sometimes be based on property values. An example of such an ordering is one that is alphabetical by authorÆs last name, or one from most recent to oldest last-modified-date. An ordering need not be based on property values, however. A professor may order a collection of course readings in

the sequence that makes sense to coordinate them with her lectures, where there is no property on the member resources that could be used to create this ordering. This set of requirements is primarily concerned with orderings that are not based on property values. The rationale for this emphasis is that property-based orderings can be obtained, even if inefficiently, using the planned search facility being developed by the DAV Searching and Locating working group. But there is no other planned WebDAV facility that could provide orderings that are not based on property values.

Another useful distinction is between server-maintained and client-maintained orderings. For server-maintained orderings, the server enforces the semantics of the ordering by placing each collection member at the appropriate position in the ordering; clients cannot alter the position of any collection mamber in the ordering. In client-maintained orderings, the client places each collection member in the ordering based on its understanding of the semantics of the ordering; the server does nothing to validate the client's positioning of the member in the ordering. These requirements address both client-maintained and server-maintained orderings.

WebDAV already provides tools that could be used for creating and maintaining ordered collections. For example, using only the base WebDAV specification, an application could create a WebDAV property called "Order" on a collection resource. The value of this property might be a list of the collection's member URIs.

What the base WebDAV specification does not do is standardize a single way to represent orderings for collections.

Different applications and services should be able to operate on

the same collection without private agreements about how to manage and examine its order. To make this possible, there needs to be a standard way to manipulate and retrieve the order of a collection, and a standard representation of the ordering.

In any situation where collaborative management of a collection takes place, and different authoring tools or WebDAV servers might be used by the collaborators, standardization is important. It is also important where a different tool may be used to view the collection from the one that was used to create it.

So for example, two users from different organizations, using different authoring tools, are working together to create a collection. One of the tools uses a property on the collection called "Order" to store an ordering of the collection. The other

tool uses a property called "SequenceNumber" on the resources identified by the collection's member URIs. If each user adds some members to the collection, there will be no reliable ordering.

3 Requirements

3.1 Shared Resources

3.1.1 A single target resource may be accessed through more than one association.

This is the primary benefit that associations bring. They allow resources to be accessed using alternative Request-URIs, and so allow them to be shared by multiple collections, on the same or different servers.

3.1.2 It is possible for clients to create additional associations that provide access to an existing resource.

It has always been possible for Web server administrators to create alternative paths to the same resource. However, clients have not had the ability to do this. Giving clients the ability to create associations makes it possible for them to create collections that share the same resources. For several scenarios motivating this requirement, see Section 2.1 above.

3.1.3 It is possible to create cross-server associations.

It is often desirable to share provide access from a collection on one server to a resource that resides on another server. For many reasons, it may be undesirable to make a physical copy of the shared resource on the local server - to conserve disk space, to respect copyright constraints, or to make any changes in the shared resources visible automatically.

3.1.4 It is possible for a client to delete an association.

It is important to note that this is a different operation from deleting the resource to which the association provides access. It must be possible to delete one association without affecting

the ability to access the same resource through other associations.

3.1.5 For any resource, it is possible to request a list of the associations that provide access to it.

This allows clients to discover what collections share the resource, thus allowing end users to navigate upward from a resource through all the collections that provide access to

it. They can use this facility to locate other related resources and to understand the contexts in which the resource is being used.

3.1.6 A down-level HTTP 1.1 or WebDAV client is able to use an association to access its target resource.

This minimal level of compatibility with older clients is needed to make deployment of WebDAV collection functionality feasible. Although new clients may be needed to create and manipulate associations, older clients should be able to read and make use of the collections built using these associations.

3.1.8 There is no requirement that associations be acyclic.

It is particularly problematic to require detection of cycles when associations cross server boundaries, but even on a single server it may be too burdensome to require detection of cycles when associations are created. In addition, there may be applications where cyclic references are desirable.

3.1.9 There may be multiple associations from the same collection to a given resource.

It is often useful to allow the same resource to be used in one collection multiple times. Typically, these are cases where the collection is ordered. Consider a case where a collection represents a book, with one internal member URI for each page in the book. A particular graphic needs to appear in several places in the book, and so there need to be multiple internal member URIs in the collection pointing to that graphic.

3.2 Ordered Collections

3.2.1 Ordering is sufficiently standardized that different applications and servers can operate on the same ordering without private agreements.

Applications and servers can apply an ordering to a collectionÆs members or discover the ordering of a collection's members without private agreements. They can also modify an ordering, at least with the help of a human user for semantics (See 3.2.3), without private agreements.

This is the minimum that is needed to support collaborative management of an ordered collection, where different authoring tools might be used by the collaborators. It is also what allows

a different tool to be used to view the collection from the one

that was used to create it. Finally, it is needed in order for servers to list collection members in order, as required by 3.2.6.

3.2.2 A collection is not required to be ordered.

A WebDAV server may support collections without supporting ordered collections. Even if the server supports ordered collections, there is no requirement that every collection on that server be ordered. Clients decide whether any given collection is ordered.

The remaining requirements apply only to collections that are ordered.

3.2.3 The semantics of an ordering are discoverable.

The semantics of an ordering define the principle or rule according to which the collection members are ordered. This principle must be discoverable if someone (or some application) other than the one that created a collection is to be able to add a member to it and determine where it makes sense to position the new member in the collection's ordering.

In some cases it may be possible for the semantics to be expressed in a machine-usable way, so that an application could automatically position a new member in the ordering. In other cases the semantics may require a human user to apply them. In either case they should be discoverable.

3.2.4 Each collection member appears in the ordering exactly once.

It would be possible to support orderings that contain only a subset of the collection members, or orderings that can contain a single collection member more than once. It is not necessary, however, since the same result can be achieved by creating a new collection with exactly the desired members, and including each member of the new collection in its ordering exactly once.

This requirement implies that the server will check, whenever a member is added to an ordering, to make sure that it is not already in the ordering. It also implies that either the protocol itself or the server will insure that whenever a new member is added to a collection, it is also added to the collection ordering.

3.2.5 An ordering does not include any resources that are not members of the collection.

The server must insure that when a member is removed from a collection, it is also removed from the collection's ordering.

3.2.6 When a client requests a listing of the members of a collection, this listing is returned in the order specified by

the collection.

This requirement frees clients from the burden of applying the ordering to the member listing. It also insures that whatever client retrieves the collection listing, the listing will appear in the same order. (An application using the listing can, of course, re-order it on the client side for its own purposes.)

3.2.7 It is possible to order the members of a collection in a client-specified way, not necessarily based on property values.

Orderings that are based on property values can be obtained by a search protocol that supports sorted result sets. This set of requirements is not concerned with such orderings. It is intended primarily to support orderings that cannot be obtained by sorting on property values.

A property is not always available that can serve as the basis for a desired ordering. For example, a professor may wish to order a collection of course readings in the sequence that coordinates the readings with her lectures. But the properties of resources at the Web site are standardized and do not include one that is appropriate to use for this purpose.

Even if the professor in the example could create a "sequencenumber" property to use in sorting the collection, this strategy would be undesirable unless she knew she would not be adding any readings or changing the order of her lectures once the values of sequencenumber were set. Inserting a new reading into the sequence would require updating the sequencenumber property of each reading that comes after the new one in the sequence. Ordered collections are intended to support this sort of case, where sorting based on a property value is impossible or inefficient.

3.2.8 It is possible for clients to discover available server-maintained orderings and to request that one of those orderings be used for a collection.

Servers may wish to make available some set of server-maintained orderings, and allow clients to choose which of them is applied to a given collection. These orderings are likely to be based on properties of the resources in a collection. For example, a server might allow resources in collections to be ordered alphabetically by author, alphabetically by title, or from most recent to earliest publication date. In order for clients to take advantage of these orderings, some way must be provided for them to discover what server-maintained orderings are available and to select one to be

applied to a given collection.

4 Acknowledgements

This draft has benefited from thoughtful discussion by Jim Amsden, Alan Babich, Steve Carter, Geoffrey Clemm, Ken Coar, Ellis Cohen, Bruce Cragun, Spencer Dawkins, Rajiv Dulepet, David Durand, Chuck Fay, Roy Fielding, Yaron Goland, Fred Hitt, Alex Hopmann, Marcus Jager, Chris Kaler, Manoj Karichainula, Rohit Khare, Daniel LaLiberte, Steve Martin, Surendra Koduru Reddy, Sam Ruby,

Nick Shelness, John Stracke, John Turner, Jim Whitehead, and others.

5 References

[URI] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax." RFC 2396. MIT/LCS, U.C. Irvine, Xerox. August, 1998.

[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.

6 Authors' Addresses

J. Slein
Xerox Corporation
800 Phillips Road
Webster, NY 14580
Email: jslein@crt.xerox.com

J. Davis
CourseNet Systems
170 Capp Street
San Francisco, CA 94110
Email: jrd3@alum.mit.edu

Expires December 18, 1999