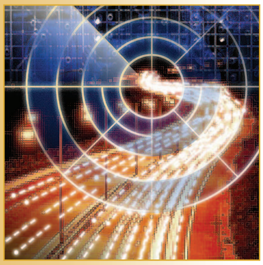

Editor: Jim Whitehead • ejw@cs.ucsc.edu

WebDAV: Versatile Collaboration Multiprotocol

The first installment of the new Standards Track explores the Web Distributed Authoring and Versioning protocol. WebDAV supports remote collaborative authoring of Web sites and individual documents, remote access to document-management systems, and more. It is the most popular network file-system protocol for use across the wide-area Internet, and it has been integrated into existing authoring tools of many types. WebDAV defines a series of extensions to the set of methods defined by HTTP version 1.1 and contains a set of features that can be used in numerous settings.

Jim Whitehead
University of California, Santa Cruz

What do Apple's iDisk and iCal, Microsoft's Outlook Express and Word, Adobe Acrobat, and Dreamweaver have in common? All use the Web Distributed Authoring and Versioning (WebDAV) protocol,[1] although in different ways and for different goals. iDisk uses WebDAV as a network file-system protocol — the primary protocol employed when Mac users access their iDisk accounts. The iCal service uses WebDAV as a calendar-access protocol. Outlook Express uses it as a mail-retrieval protocol when interacting with the Hotmail service. Word uses it to enable remote collaborative authoring of individual documents, and Acrobat uses it to allow shared commenting and annotation on PDF documents. Finally, Dreamweaver uses WebDAV to support collaborative work on entire Web sites. Clearly, this versatile protocol has multiple uses and strengths.

In 1996, the IETF's WebDAV working group began work to develop interoperability protocols for remote Web site authoring. Three years later, the WG achieved a major milestone in publishing the WebDAV protocol, thus clearing the way for commercial and open-source adoption. During the development process, the WebDAV WG tailored the protocol to support remote collaborative authoring of Web sites and individual documents, as well as remote access to document-management systems. Yet, even the first adopters used WebDAV for other purposes.

Today, WebDAV is the most popular network file-system protocol for use across the wide-area Internet (the Network File System and Andrew File System are more frequently used behind firewalls). WebDAV is also the third most widely used email-retrieval protocol (behind POP and IMAP), driven entirely

1089-7801/05/$20.00 © 2005 IEEE

WebDAV Collaboration Scenarios

With its compact yet powerful capabilities, the WebDAV protocol supports many forms of collaborative work beyond its initial focus on Web site authoring. The following scenarios detail WebDAV's versatility, highlighting multiple ways in which the protocol supports collaboration.

Figure A shows a collaborative-writing scenario in which three collaborators at three different sites jointly author a document using Microsoft Word's WebDAV capabilities. Word uses the WebDAV protocol to interact with the shared document, which is stored on a WebDAV server in the Los Angeles office.

In Figure B, the shared-commenting scenario, two collaborators use Adobe Acrobat's facilities to share comments and annotations on a journal article located in the IEEE Xplore digital library. Annotations are stored separately from the document on a WebDAV server that isn't part of IEEE Xplore. Annotations are visible only to the two collaborators; other users downloading the same article can't see them.

Figure C illustrates the file-sharing scenario. Three employees in different branch offices of a company exchange files by dragging them from their desktops to and from a WebDAV server. They can reliably exchange large files while avoiding their email system's attachment-size limits. Each collaborator works with a different operating system and its native WebDAV file-system integration (Web Folders for Windows, davfs for Linux, and Finder on the Mac).

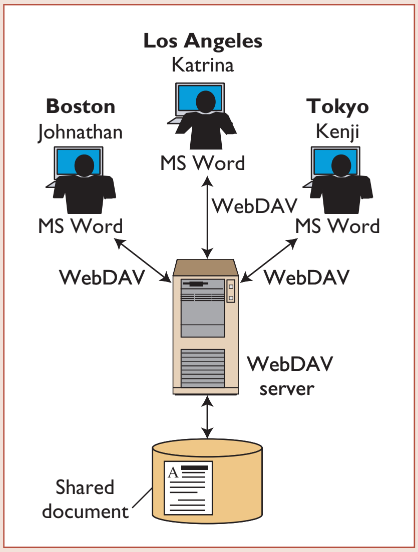

Figure A. Collaborative-writing scenario. Three authors write collaboratively using Word and WebDAV.

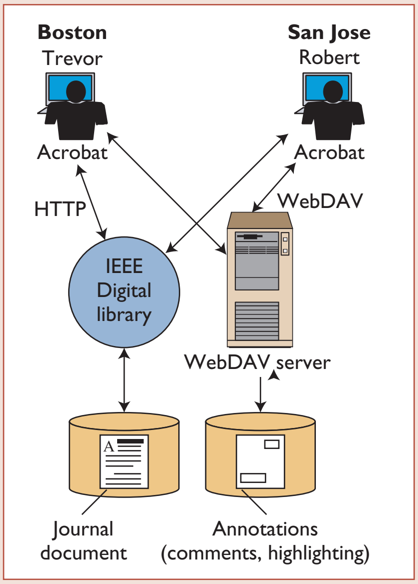

Figure B. Shared-commenting scenario. Authors can annotate a shared document via Acrobat and WebDAV.

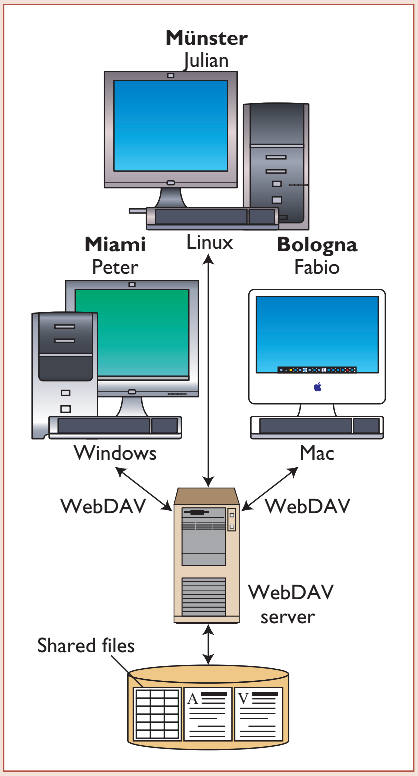

Figure C. File-sharing scenario. Three people share files using desktop drag-and-drop into a WebDAV-server-mapped disk drive.

by Outlook, Outlook Express, and the Exchange mail server. With iCal and Mozilla Calendar using it, WebDAV is now the de facto open standard for Internet calendar access and sharing. It is also the primary protocol supporting its core functions: remote collaborative document authoring, annotation, and Web site authoring.

At its core, the WebDAV protocol has three powerful features: overwrite prevention, property (metadata) management, and namespace control. Recent work in the WebDAV WG has focused on extending the protocol's capabilities to support access control and ordered collections. In this article, the first in *IC*'s Standards Track series, I'll describe the WebDAV protocol and its recent extensions, the ongoing work in the WebDAV WG, and possibilities for future capabilities. (In the interest of space, I won't cover the DeltaV protocol,[2] which adds versioning and configuration-management capabilities, but a detailed introduction is available elsewhere.[3])

WebDAV Protocol Functionality

The WebDAV protocol is designed to be integrated into existing authoring tools, adding Web-based remote authoring capabilities to the tools users already know. Just as Web browser users are largely unaware of the HTTP network traffic that requests and downloads Web pages, users of WebDAV-enabled authoring tools are largely unaware of the protocol's use.

WebDAV extends the methods defined by HTTP version 1.1 (`get`, `head`, `post`, `options`, `put`, `delete`, and `trace`). The WebDAV protocol contains a set of features that can be used in numerous settings by applications that support collaborative work on remotely authored documents. We can partition WebDAV's seven methods, as HTTP operations are called, into three groups: overwrite protection (`lock`, `unlock`), metadata management (`propfind`, `proppatch`), and namespace management (`copy`, `move`, `mkcol`).

Overwrite Prevention

Once two or more people start collaborating on a document, the issue of write control comes to the fore. If everyone can write to the same, unversioned document, contributors' changes can be lost as one collaborator after another writes changes without first folding in previous updates.

Several common techniques can help alleviate this "lost update" problem.

- *Passing the baton or edit token.* In this scheme, collaborators agree to a social convention in which they communicate when one author finishes working, and it's safe for another to begin. The active author controls access by sending messages that grab or release the edit token. Following the analogy of relay race runners passing a baton, the edit token is an imaginary item passed among authors, representing shared understanding of who can currently edit the file.
- *Shared locks* (also known as advisory locks or reservations). An author submits a modification-intention message to the computer that controls access to the document, which records this author's intent to edit. If another author similarly indicates intent to edit, the computer announces that the document is currently being edited. However, the second author can still edit the document if desired—presumably, after contacting the other author to negotiate access or taking advantage of extra-system knowledge that no conflict will result (for example, the other author is in a meeting).
- *Exclusive locking.* An author indicates the intent to modify a given document, and the computer that controls access to the document responds by locking it. Once the document is locked, only the lock owner can modify it. The computer refuses other authors who try to edit the document.

These schemes vary from least protective and most flexible (baton passing) to most protective and least flexible (exclusive locking).

Currently, the WebDAV approach provides facilities for both shared and exclusive locking (no additional protocol support is required for baton passing). Authoring programs, acting on behalf of users, request both kinds of lock using the `lock` method, and remove them using `unlock`. This dual lock support provides sufficiently flexible locks to accommodate a wide range of collaborations. Whereas shared locks best support collaborators with rich awareness of each other's activities, exclusive locks provide a more stringent guarantee of conflict avoidance for less aware collaborators or for periods of high contention for a document.

A lock can cover a single resource, including all its non-live (or static) properties, or a hierarchy of resources (for example, a collection and all its member resources). WebDAV's `lockdiscovery` property allows authors to find out if any locks exist on a given Web resource. (There is no concept of a read lock because the Web's design requires no locks for reading Web pages.) Most WebDAV applications that incorporate locks just use exclusive ones.

Metadata Management

Several goals motivate WebDAV's metadata facilities.[4] The ability to associate metadata with resources is valuable for a wide range of document-management activities, such as recording workflow state and tailoring the system for specific document-processing applications. Version-control systems invariably associate metadata with revisions to represent revision identifiers, comments, predecessor and successor relationships, and revision labels. Providing access to typical file-system metadata, such as the creation date, size, and last-modified date, is also desirable.

Metadata items in WebDAV are known as *properties*, which are name-value pairs. The name comprises a namespace name (a URL or URI, equivalent to an XML namespace name) and a property name. The value is a well-formed sequence of XML content. If, for instance, a property namespace is a URL, the party defining the property can give it uniqueness without central registration by using URLs chosen from within a domain whose name the party controls. For example, a company that controls a given domain name — say, "widgets.com" — can use "http://widgets.com/" as the namespace name, and then assign property names within the namespace, such as "properties/color/."

As an example, consider the WebDAV protocol's `DAV:getcontentlength` property, which is attached to every WebDAV resource. It gives the length, in bytes, of the response generated by a `get` on its resource. The property namespace is the URI "`DAV:`," which is reserved for use by WebDAV. The property's name is "`getcontentlength`." A sample value of this property is

```
Namespace: DAV:
Property: getcontentlength
Value: <D:getcontentlength
xmlns:D="DAV:"> 3422
</D:getcontentlength>
```

By convention, the enclosing XML element for a WebDAV property encodes the property's namespace and property name. In this case, the length, 3,422 bytes, is enclosed within the `<D:getcontentlength>` XML element.

WebDAV properties can be either `dead` or `live`. A dead property is one for which the client maintains syntax, semantics, and consistency, while the server performs little, if any, processing on the data. These properties are set and updated by client applications. In contrast, a live property is one for which the server provides the value. For example, the server computes the resource's length to supply the value for the `DAV:getcontentlength` property. A live property also accommodates the case in which the client provides the value, and the server performs syntax and consistency checks on it. In essence, a live property is one for which the server performs a computation associated with setting or retrieving its value, whereas a dead property requires no computations other than checking that the XML is well-formed.

WebDAV provides two methods for interacting with properties: `propfind` and `proppatch`. Applications use `propfind` to retrieve values for one or more properties, from one or more resources; these values are wrapped in a `multistatus` XML element, which allows responses from multiple resources to be combined into a single XML sequence that can be transmitted in response to a single HTTP method invocation. For example, a client can request a listing of all resources in a collection hierarchy using `propfind`; the server reports the results in a single `multistatus response`. The `proppatch` method allows clients to add or remove one or more properties from a resource by submitting a series of `add/remove` operations. `Proppatch`'s operation is atomic — that is, it performs either all `add/remove` operations or none of them.

Namespace Management

Hierarchical organization structures are the most common way to organize files in current operating systems. WebDAV provides support for creating collections (directories) of Web resources (files) via the `mkcol` method. Applications generate collection-membership lists using the `propfind` method with a `depth` of 1 (a collection and all its members), requesting file-system-like metadata such as `DAV:lastmodified` and `DAV:getcontentlenth`. The `propfind` response lists all collection members and their requested metadata, which can be formatted into a directory listing and displayed to a user. Applications can copy or move resources between collections using the `copy (move)` method with `depth 0` for a single resource or `depth infinity` for an entire collection hierarchy.

Two recent extensions to the WebDAV protocol provide improved access control, as well as the ability to maintain persistent orderings for the members of collections.

Access Control

All but the simplest HTTP servers have mechanisms for controlling access to the resources they host, enabling administrators to control which resources are readable, writeable, and by whom. Given that WebDAV servers are HTTP servers, they inherit these HTTP access control mechanisms.

Administrators use different mechanisms on each server to control resource access. Apache's configuration files, for example, must be modified in conjunction with file-system permissions, and no other server uses the Apache configuration file format. Unfortunately, every server has a different configuration mechanism, and they typically aren't remotely accessible. Those that are remotely accessible generally take the form of Web applications aimed at human users, providing no support for programmatic access. These twin drawbacks — the lack of both a standard access control configuration mechanism and program-

WebDAV Method Quick Reference

Given that WebDAV builds on top of HTTP, and that there are multiple extensions to it, keeping track of the existing, new, and proposed HTTP methods can be tricky. Table A includes known HTTP and WebDAV methods, sorted by their points of origin. In addition to the methods listed here, new protocols often introduce protocol-specific properties that encode some of the protocols' behavior, as well as modifications to the behavior of existing methods, such as `options`.

Table A. HTTP and WebDAV methods.

Method	Action
HTTP (RFC 2616)	
GET	Retrieve a representation of a resource (read operation)
HEAD	Return HTTP headers, but not content, as if doing a GET (read metadata)
PUT	Write a resource
DELETE	Make a resource inaccessible via the specified URL
POST	Submit a Web form, tunnel other protocols
OPTIONS	Perform resource discovery (list methods the resource supports)
TRACE	Repeat received message (diagnostic or development use)
CONNECT	For use by proxies to dynamically switch to being a (SSL) tunnel
WebDAV (RFC 2518)	
PROPFIND	Retrieve resource properties; list members of a collection
PROPPATCH	Write resource properties
LOCK	Lock a resource, or collection of resources, with a shared or exclusive lock
UNLOCK	Remove a lock
MKCOL	Create a new collection
COPY	Copy a resource or collection hierarchy
MOVE	Move a resource or collection hierarchy
Access Control (RFC 3744)	
ACL	Write the access control list on a resource
Ordered Collections (RFC 3648)	
ORDERPATCH	Modify the in-collection ordering of a resource
Bindings (draft-ietf-webdav-bind)	
BIND	Make an existing resource a member of an existing collection (create hard link)
UNBIND	Remove containment relationship between a resource and a collection
REBIND	Atomically move a resource from one collection to another (change source of a containment relationship)
Redirect References (draft-ietf-webdav-redirectref-protocol)	
MKREDIRECTREF	Create a redirect reference resource
UPDATEREDIRECTREF	Cause a redirect reference to point to another URL
WebDAV Search (draft-reschke-webdav-search)	
SEARCH	Search the properties and content of a resource or hierarchy

matic access — motivated the WebDAV Access Control Protocol's development.[5]

The WebDAV Access Control Protocol extends the core WebDAV protocol to provide an interoperable way to control access to content and metadata managed by WebDAV servers. It provides discretionary access control, which lets resource owners and administrators control the operations other users can perform on the resource.

Principals

A *principal* is a person, computational agent (such as a Web crawler), or a group of these. To manage Web site access, an administrator maps principals to permissible operations in access control lists (ACLs, pronounced "ack-ulls"), which contain a series of access control entries (ACEs). An ACE lists a principal and a series of privileges that are granted or denied to them. Every resource has a single ACL associated with it, and the server evaluates the requesting principal against the ACL before performing all requested operations.

Principals can be collected into groups, representing organizational units such as departments or committee members. The `group-member-set` property (defined only on group principals) lists the members of a group, and the `group-membership` property (defined on all principals) lists all the groups to which a given principal belongs.

Administrators maintaining ACLs need to see human-readable names for the principals they're granting or denying privileges — something the principal's identifying URL doesn't provide. The `displayname` property carries this information. Because principal information will likely be maintained in a directory, with only portions exported for use by the access control protocol, the ACL protocol provides the `alternate-URI-set` property, defined on all principals. This property exposes one or more URLs that point to additional places where a client could find more information about the principal, such as an LDAP scheme URL, which identifies a directory entry.

To find a given principal by name — an important ability, given that a server can have many principals — the access control protocol uses the flexible `report` method, which supports a wide range of focused queries. Specific search semantics are associated with each report type associated with this method. The `principal-property-search` report searches all principal resources on a server to locate those with property values that match the search specification. The `principal-property-search` report finds principals whose `displayname` property matches the search name. The `acl-principal-prop-set` report is also useful for returning human-readable names for all the principals in an ACL.

Privileges

Privileges are used in ACEs to control access to resources. Each represents one or more HTTP operations. The `read` privilege, for example, controls methods that read a resource's body or properties, thus affecting the `get` and `propfind` methods. The `write` privilege controls methods that lock the resource or modify the content and dead properties, such as `put` and `proppatch`. With a collection, the `write` privilege also controls the membership — that is, adding resources to or deleting them from the collection.

One challenge in drafting the access control protocol was to develop a privilege set flexible enough to faithfully expose the underlying repository's privilege model, yet standard enough to ensure interoperability. Consider, for example, a WebDAV server implemented on top of a repository that supports only a `write-all` privilege that gives authorized users write access to the resource's primary content (`get` response body) and properties. The access control protocol includes privileges called `write-content` and `write-properties`, which control the modification of the resource body and properties, respectively. For this repository, the implementation would need to map both of these to the repository's `write-all` privilege; as an undesirable side effect, the protocol's `write-properties` privilege would grant write access to both the properties and the primary content. One solution would have been to leave the separate `write-properties` privilege out of the protocol, thus ensuring a clear mapping between the protocol and this repository. Yet, having a separate `write-properties` privilege makes it possible to annotate a resource (in a property) without granting write access to it, a desirable feature. Additionally, other repositories might have both `write-content` and `write-properties` privileges, and the protocol needed to be able to expose this flexibility.

To address this issue, the access control protocol introduced two privilege types:

- `Abstract` privileges can't be granted or denied in an ACE.
- `Aggregate` privileges group together sets of related privileges, such that granting or denying the privilege effectively grants or denies all its contained privileges.

What's important is how abstract and aggregate privileges work together. When the underlying repository implements only a `write-all` privilege, a server implementer could declare the protocol's `write-properties` privilege to be abstract, thus informing clients that they can't directly set write-properties in an ACE. The write-properties privilege then becomes a member of the aggregate write privilege. The server implementer would treat the write-content privilege similarly, also making it abstract and a member of the aggregate write privilege. The net result for this repository is that a client could use the write privilege only in ACEs, rather than using write-content or write-properties directly. A repository that gives separate privileges for writing properties and content could use the same aggregation hierarchy, instead exposing write-content and write-properties as normal (rather than abstract) privileges. In this case, a client could use write, write-content, or write-properties in an ACE.

The access control protocol defines the following privilege set:

- Read controls the methods that return the resource's state (body and properties).
- Read-acl controls reading the ACL on a resource.
- Read-current-user-privilege-set controls who can read the current privilege set granted to the current user on the resource.
- Write controls methods that write the resource's state (body and properties).
- Write-properties controls methods that write the resource's properties (proppatch).
- Write-content controls methods that write the resource body (put)
- Write-acl controls the writing of the ACL on a resource (ACL method).
- Bind controls adding a new member to a collection, as when creating a new resource via put or a new collection via mkcol.
- Unbind controls the removal of a member from a collection (delete).
- Unlock controls the unlocking of a locked resource by principals other than the lock owner, who can always unlock a locked resource.
- All is an aggregate that contains the entire privilege set that can be applied to the resource.

Given that privileges, ACLs, and ACEs provide the vocabulary for controlling access to resources, it is no surprise that they present the most complex aspects of the Access Control Protocol. In comparison, the mechanism for reading and writing ACLs is straightforward.

Working with ACLs

A resource's defined ACL is found in the acl property, which returns the list in XML format. The ACL includes a series of ACEs that each specify a principal, a list of privileges, and whether they're granted or denied to the principal. The read-acl privilege controls read access for this property. The acl method supports only the writing of an entire ACL; it requires clients to first retrieve the existing list, make modifications, and then resubmit the new list in its entirety. The write-acl privilege controls whether the acl method can be used to modify the ACL on a resource.

The WebDAV Access Control Protocol was published in May 2004 and is currently implemented by SAP Netweaver, Xythos WebFile Server and Oracle XML DB, on the server side, and by Xythos WebFile on the client side.

Ordered Collections

In some collections, it's desirable for the member resources to be listed always in the same order. We would want a collection that included the chapters of a book to return the list of chapters in order, for example, rather than placing "chapter10" right after "chapter1" as an alphabetical sort would. A user-maintained ordering could achieve the desired sequence in this case.

The WebDAV Ordered Collections Protocol, approved in December 2003, provides a mechanism for creating this kind of persistent, user-maintained ordering of the members of a collection.[6] The protocol has three main capabilities:

- creating a new ordered collection,
- creating or moving resources in ordered collections, and
- changing the ordering of ordered collection members.

The mkcol method creates ordered collections, just as it creates unordered collections. When present, the new ordering-type HTTP header indicates that the collection is to be ordered and gives an identifier for the type of ordering to be maintained.

When creating a new resource in an ordered collection using put or performing a move or copy whose destination is in an ordered collection, the client must specify where the resource belongs in the ordering. The new position HTTP header works with put, move, and copy to specify whether the resource is first, last, or comes before or after an existing member.

Learning More

Among the wealth of information available on WebDAV, the following list includes a few highlights. The first two sites are useful for finding protocol specifications, and Dusseault's book is the most comprehensive, compact reference on WebDAV currently available.

- The WebDAV Resources site (www.webdav.org) includes WebDAV news, specifications, and open-source projects.
- Julian Reschke scrupulously maintains a list of in-progress WebDAV specifications at the Greenbytes WebDAV Resources site (www.greenbytes.de/tech/webdav/).
- Wikipedia's provides a nice, brief introduction to WebDAV and a list of some WebDAV projects (http://en.wikipedia.org/wiki/WebDAV).
- The Sharemation site (www.sharemation.com), operated by Xythos, provides free WebDAV-enabled accounts for evaluation use.
- Lisa Dusseault's book, *WebDAV: Next-Generation Collaborative Web Authoring* (Prentice-Hall, 2004), provides a detailed description of the WebDAV, Access Control, and DeltaV protocols, along with some design rationale.

The new `orderpatch` method changes the ordering of one or more collection members, providing an ordering specification for each resource being reordered.

As with unordered collections, clients can list the members of an ordered collection in order using the `propfind` method.

What's Next

The WebDAV development community is still working to complete several major protocol efforts. The IETF WebDAV WG is officially working on four of these: Quota, Bindings, Redirect References, and revisions to the WebDAV Protocol (known as 2518bis).

The Quota Protocol lets administrators read, write, and enforce space quotas on WebDAV servers.[7] A space quota sets an upper limit on the amount of storage space a principal can use on a server; it is especially useful for servers with large and diverse user populations, such as the student body of a university.

The Bindings protocol permits the direct creation and removal of containment relationships between collections and other resources without modifying other collection memberships.[8] This facility is similar to the notion of a "hard" link in an operating system.

Redirect references instruct Web servers to redirect requests to other locations using HTTP 3xx responses (Redirection). When moving a Web page to a new URL, the author should leave a forwarding address to direct visitors from the original location to the new URL. In HTTP, this forwarding address is communicated using a 301 or 302 response to a `get` request. This protocol makes it possible to remotely author resources whose purpose is to respond to `get` requests with a 3xx response code,[9] thereby leaving forwarding address information for moved Web pages.

Following the IETF's standards track — in which protocols progress from "proposed" to "draft" and, finally, to "standard," with revisions between these milestones — the WebDAV WG is also incorporating implementation feedback and fixing technical and editorial errors with the revised WebDAV Protocol, known as 2518bis.[10] The "bis" notation indicates that this is a revision of the original protocol document (RFC 2518). The suffix is frequently used in IETF specification revisions (for example, LDAP v3 bis), perhaps coming from the musical notation directing a phrase or passage to be repeated.

Julian Reschke of Greenbytes Software has also been pushing forward the WebDAV Search and Property Datatypes specifications, among several other contributions to working group activities.

In addition to these efforts, work continues on WebDAV Search (DASL), which would let users search WebDAV servers for resources that match SQL-like search expressions.[11] Most searches are over WebDAV property values, but DASL also allows searches of resource contents.

Because they're represented on the wire as sequences of XML, WebDAV properties don't have associated type information. Yet, many WebDAV servers use underlying repositories, such as relational databases, that can provide significantly improved storage and searching when property data types are known. To address this fact, Reschke is pushing Property Datatypes, which allows property values to be typed using `proppatch` and to be retrieved via `propfind`.[12]

Predicting which specification will be completed next is difficult, but the Quota and Property Datatypes drafts are currently solid specifications with multiple implementations — always a good sign of maturity. Despite its immense utility and multiple existing implementations, WebDAV Search isn't as poised for rapid completion due to the specification's complexity and its partial dependence on the Property Datatypes effort. Redirect References, Bindings, and 2518bis are all far along, but they'll require moderate effort to bring to completion. Like many IETF protocol efforts, additional workers would help speed development, but these drafts are steadily progressing as all have seen new revisions in the past eight months.

Many items remain on the WebDAV wish list. Access to WebDAV server data from within HTML would be a plus, as would including property values in displayed Web pages. WebDAV could also play an important role with cameras and other data-collection devices. As wireless networking becomes more ubiquitous and the cost of wireless chipsets drop, wireless networking will increasingly be standard in digital cameras, portable music players, and various scientific data loggers. With its ability to record metadata along with content, leverage HTTP, and provide rich access control, WebDAV seems a natural protocol in this space. Photographers could immediately send digital photos to remote WebDAV servers as soon as they took them, while portable music players could use WebDAV to synchronize with remote music libraries.

The Calendaring and Scheduling Consortium (www.calconnect.org) is developing a new standard called CalDAV[13] for WebDAV's increasingly important use in calendaring. CalDAV builds on WebDAV to offer calendaring and scheduling operations. It defines calendar-specific privileges, for use with the access control protocol, and mappings of calendar information into WebDAV properties. This effort, when completed, will allow individuals to share their calendars over the Internet, easily scheduling meetings both within and across organizational boundaries. This capability is especially useful for workers who interact with people from a wide range of organizations (such as service techs and sales people) and for families that want to synchronize schedules among family members, whether from home, work, or school.

Finally, space itself might be the final frontier for WebDAV. In simulated Mars missions today, the Mars Society uses the WebDAV protocol (via Windows Web Folders) to transfer data collected from field missions back to a central machine at Mars Society Headquarters. Perhaps future Mars missions will send photographs and datasets back to Earth via WebDAV over Interplanetary IP.

References

1. Y. Goland et al., "HTTP Extensions for Distributed Authoring – WebDAV," Internet Proposed Standard, RFC 2518, Feb. 1999.
2. G. Clemm, J. Amsden, T. Ellison, C. Kaler, J. Whitehead, "Versioning Extensions to WebDAV." Internet Proposed Standard RFC 3253, Mar. 2002.
3. L. Dusseault, *WebDAV: Next-Generation Collaborative Web Authoring*, Prentice-Hall, 2004.
4. E.J. Whitehead Jr. and Y.Y. Goland, "The WebDAV Property Design," *Software, Practice and Experience*, vol. 34, 2004, pp 135–161.
5. G. Clemm et al., "Web Distributed Authoring and Versioning (WebDAV) Access Control Protocol," Internet Proposed Standard RFC 3744, May 2004.
6. J. Whitehead and J. Reschke, "Web Distributed Authoring and Versioning (WebDAV) Ordered Collections Protocol," Internet Proposed Standard RFC 3648, Dec. 2003.
7. B. Korver and L. Dusseault, "Quota and Size Properties for DAV Collections," IETF Internet draft, Jul. 2004; work in progress.
8. G. Clemm et al., "Binding Extensions to Web Distributed Authoring and Versioning (WebDAV)," IETF Internet draft, Sept. 2004; work in progress.
9. J. Whitehead, G. Clemm, and J.F. Reschke, "Web Distributed Authoring and Versioning (WebDAV) Redirect Reference Resources," IETF Internet draft, Oct. 2004; work in progress.
10. L. Dusseault and J. Crawford, "HTTP Extensions for Distributed Authoring - WebDAV RFC2518 bis," IETF Internet draft, 17 July 2004; work-in-progress.
11. J.F. Reschke et al., "WebDAV Search," IETF Internet draft, Sept. 2004; work in progress.
12. J.F. Reschke, "Datatypes for WebDAV Properties," IETF Internet draft, Sept. 2004; work in progress.
13. C. Daboo, B. Desruisseaux, and L. Dusseault, "Calendaring and Scheduling Extensions to WebDAV (CalDAV)," IETF Internet draft, 20 Sept. 2004; work in progress.

Jim Whitehead is an assistant professor in the Department of Computer Science at the University of California, Santa Cruz. He founded the IETF WebDAV Working Group and served as its chair from inception in March 1997 through March 2004. His research interests include collaborative authoring, software configuration management, software evolution, and Web engineering. Whitehead received a PhD in information and computer science from the University of California, Irvine. He is a member of the ACM, Usenix, and the IEEE. Contact him at ejw@cs.ucsc.edu.