

The Web as a Collaborative, Writeable Medium



An Introduction to the IETF WebDAV
Standard

Jim Whitehead

Univ. of California, Irvine

ejw@ics.uci.edu

What is WEBDAV?



**Working Group on Distributed Authoring and
Versioning on the World Wide Web**

**Goal: To enable distributed web authoring
tools to be broadly interoperable.**

Home page:

<http://www.ics.uci.edu/pub/ietf/webdav/>

WebDAV: Extending HTTP



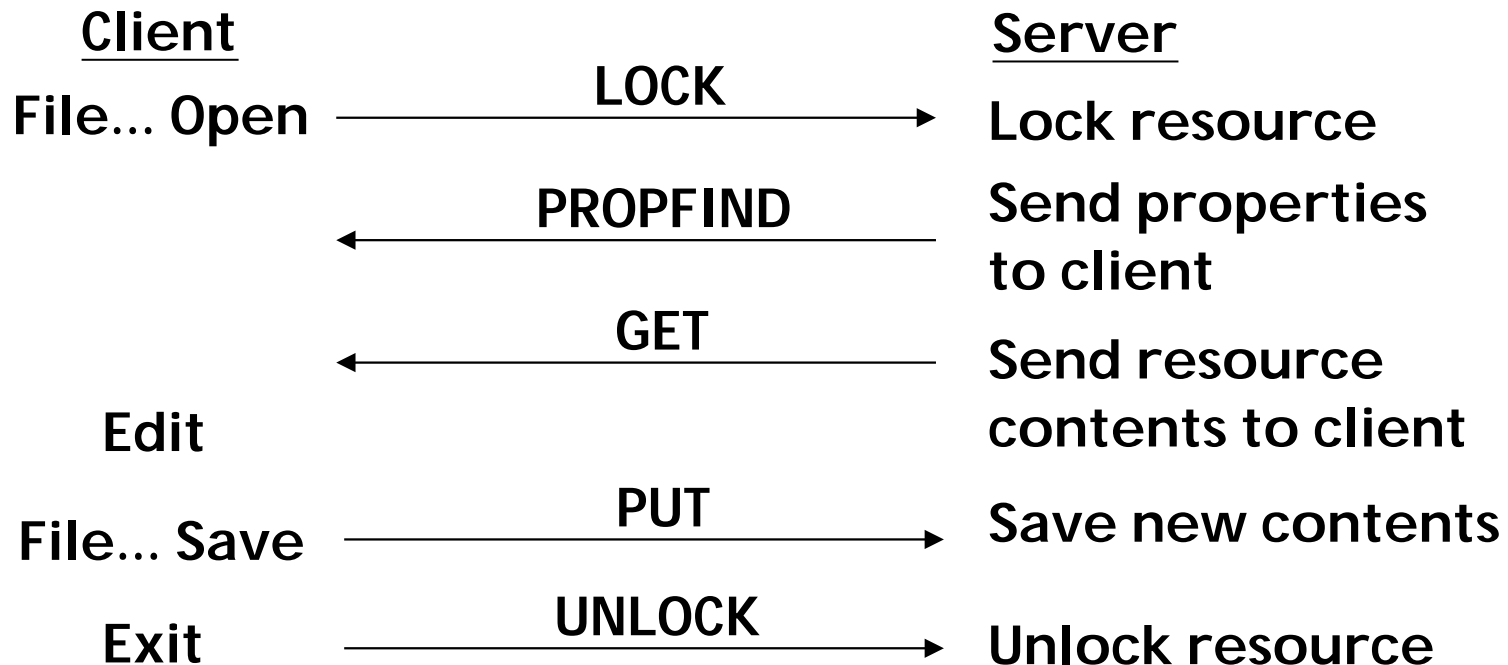
- **WebDAV is a major extension to HTTP**
 - WebDAV adds properties and collections to the HTTP data model
- **WebDAV provides facilities for**
 - Properties - list, add, remove
 - Namespace Operations - move, copy
 - Overwrite prevention - lock, unlock
 - Collections - mkcol, hierarchy operations

Using WebDAV



- **You have a report to produce**
 - Fire up your favorite word processor & start editing
- **You decide to bring on another author**
 - Using the same word processor, save to the Web
 - Start collaboration on the document by editing in-place on the Web
- **A seamless transition from individual to collaborative work**

Application Use of WebDAV



Visions for WebDAV

- **Participants in WebDAV have many views on what it is:**
 - **A protocol for collaborative authoring**
 - **A large-grain Web-based network file system, with nice high-latency behavior.**
 - **A data integration technology for accessing a wide range of repositories**
 - **document mgmt. systems, configuration mgmt. systems, filesystems, etc.**
 - **A protocol for remote software engineering**
- **All views are correct!**

Facets of WEBDAV



- **There are many ways to view the DAV work:**
 - **Collaboration infrastructure**
 - **Metadata recording infrastructure**
 - **Namespace management infrastructure**
 - **Ordered collection infrastructure**
 - **Versioning infrastructure**
 - **Access control infrastructure**
 - **Searching infrastructure - DASL**

Collaboration Infrastructure



- **Whole resource locking supports:**
 - remote collaborative authoring of HTML pages and associated images
 - remote collaborative authoring of any media type (word processing, presentations, etc.)
- **Infrastructure for development of asynchronous, widely distributed, hypertext-aware, collaborative editing tools.**

Metadata Recording Infrastructure



- **Metadata support**
 - Properties. (name, value) pairs can be created, modified, deleted, and read on Web resources.
 - Consistency of properties can be maintained by the server or the client
 - Property values are well-formed Extensible Markup Language (XML)
- **Infrastructure for how to record information about Web data**

Namespace Management Infrastructure



- **Remote name space management:**
 - Copy and Move individual resources, and hierarchies of resource
 - Create and modify (ordered) collections of resources
 - Add/remove members by-reference
- **Infrastructure for remotely organizing and viewing collections of Web resources**

Versioning Infrastructure



- **Versioning is a key part of WebDAV**
 - check-out, check-in
 - version graph history
 - comments on check-out/check-in
 - browse old versions
 - automatic versioning for unaware clients
 - basic, high-value configuration management operations
- **Infrastructure for remotely versioning Web resources**

Access Control Infrastructure



■ Access Control:

- The ability to remotely control who can read and write a resource
- Key challenge:
 - Expose the access control capabilities of the repository...
 - ...while ensuring the client-side user interface can be simple (I.e., avoid lots of feature discovery)
- Will be the focus of a new working group.

■ Infrastructure for remotely creating collaboration groups

Searching Infrastructure



- **Searching a WebDAV repository - DASL:**
 - Search for resources with a given property, or a given property value
 - Search for a substring inside a resource body
 - Search scope can be one resource, a collection of resources, a hierarchy of resources, or a whole server
- **Infrastructure for remote searching**

Document Roadmap



WebDAV Working Group:

Distributed Authoring

Locking, Properties, Copy/Move
RFC 2518 complete

Versioning

Checkin/Checkout, Variants
Finish: Mid 2000, as new WG

Ordered Collections

Requirements and protocol for
ordered collections, external members
Finish: June/July 1999

Access Control

Requirements and protocol
for remote access control
Finish: Mid 2000, as new WG

DASL Working Group:

Searching

Requirements and protocol for
searching a WebDAV repository
Finish: Fall 1999

Future work:

Notifications


Requirements and protocol for
asynchronous notifications

Commercial WebDAV Products



- **Several companies will be supporting WebDAV in their upcoming products:**
 - **Microsoft**
 - | Internet Information Services 5 (IIS)
 - | Internet Explorer 5
 - | Office 2000
 - **Novell**
 - | GroupWise Web Publisher
 - **Xerox**
 - | DocuShare 2.0

Commercial WebDAV Products (2)



- **Commercial products:**

- **Cybertteams**

- | Website Director (integrated WebDAV server and workflow system)

- **DataChannel**

- | RIO

- **IBM**

- | WebSphere (rumored)

- **Complete list:**

- | <http://www.webdav.org/projects/>

Open Source Projects



- **WebDAV has picked up strong support in the Open Source community as well:**
 - **Greg Stein's mod_dav Apache module**
 - **Joe Orton's sitecopy, site synchronization tool**
 - **Jim Davis, PyDAV, Python-based WebDAV server**
 - **Steve Ball and Daniel Lopez, TcIDAV, Tcl API for WebDAV**
 - **UCIs WebDAV Explorer, DAV Posties**

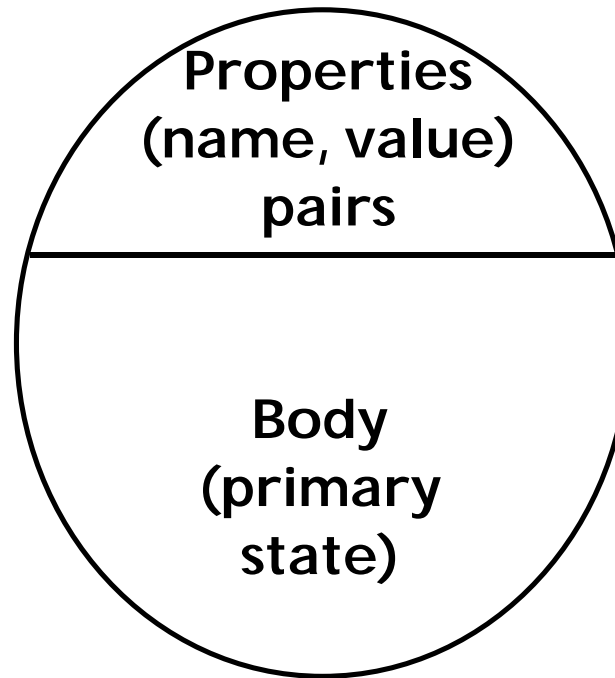
WebDAV Distributed Authoring Protocol



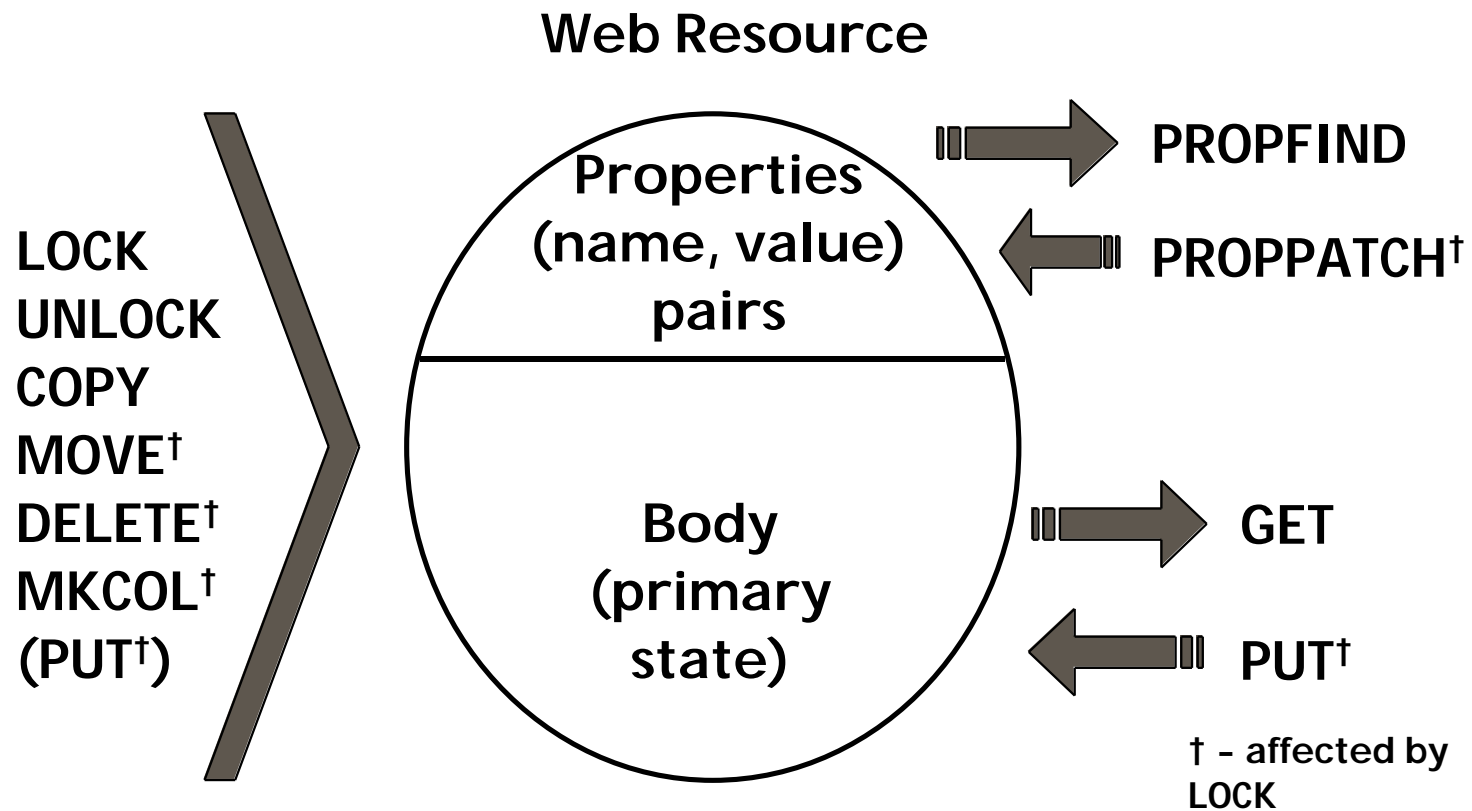
Object Model and Design
Overview

WebDAV Object Model

Web Resource



Scope of WebDAV Methods





Properties

Properties: Naming



- **Properties are (name, value) pairs**
- **Property names are URIs**
 - can be a URL (no registration needed)
 - can be a URI (register new URI scheme)
- **Benefits:**
 - Due to ownership of a domain name, URLs provide globally unique names without registration
 - URLs allow rapid development and deployment of new schemas
 - Stable, long-lifetime schemas can be named with a URI scheme, which is registered with IANA


Properties:

Name/Instance Distinction

- A property name URI names the syntax and semantics of the property.
 - Only one instance of a property may be created on a resource (but may be multi-valued)
 - “live” properties: server maintains consistency by enforcing syntax and semantics
 - non-live properties: client enforces syntax and semantics (property may be inconsistent)
- Benefits:
 - server can provide properties with values it generates
 - client can define new properties unknown to the server
 - major benefit: flexibility

Properties:

Value is well-formed XML



- **The value of a property is a well-formed XML (eXtensible Markup Language) fragment**
 - WebDAV also requires use of XML Namespaces
- **Benefits:**
 - extensibility: namespaces allow mix of elements in properties
 - i18n support: XML supports ISO 10646 encoding of characters
 - property contents are structured values
 - supports “plug-and-play” of W3C RDF work

Properties: PROPPATCH

- **PROPPATCH method is used to create and remove properties from a resource**
 - Property creation and removal directives are specified with XML "create" and "remove" elements
 - Directives are executed sequentially and atomically
- **Benefits:**
 - simple method handles modification to the state of a property
 - can modify several properties at once, with one network round trip
 - atomicity ensures properties will not be left in an inconsistent state

Properties: PROPFIND



- **PROPFIND retrieves properties from a resource.**
 - Retrieve all property names and values
 - Retrieve only specified names and values
 - Retrieve only a list of property names
- **Benefits:**
 - can retrieve property information with a single network round trip

PROPFIND, implicit allprop



PROPFIND /demo.txt HTTP/1.1

Host: dav.ics.uci.edu

Content-Length: 0

HTTP/1.1 207 Multi-Status

Server: DAV-demo-server/1.0

Date: Tue, 09 Feb 1999 00:25:47 GMT

Content-Type: text/xml; charset="utf-8"

Content-Length: 891

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

```
<a:multistatus xmlns:a="DAV:">
```

```
<a:response>
```

```
  <a:href>http://dav.ics.uci.edu/demo.txt</a:href>
```

```
  <a:propstat>
```

PROPFIND, implicit allprop (2)

```
<a:status>HTTP/1.1 200 OK</a:status>
  <a:prop>
    <a:getcontentlength>59</a:getcontentlength>
    <a:creationdate>
      1999-02-09T00:16:40.574Z
    </a:creationdate>
    <a:displayname>demo.txt</a:displayname>
    <a:getetag>"10598798c153be1:a82"</a:getetag>
    <a:getlastmodified>
      Tue, 09 Feb 1999 00:17:36 GMT
    </a:getlastmodified>
    <a:resourcetype/>
    <a:lockdiscovery/>
```

PROPFIND, implicit allprop (3)

```
<a:supportedlock>
  <a:lockentry>
    <a:locktype><a:write/></a:locktype>
    <a:lockscope><a:shared/></a:lockscope>
  </a:lockentry>
  <a:lockentry>
    <a:locktype><a:write/></a:locktype>
    <a:lockscope><a:exclusive/></a:lockscope>
  </a:lockentry>
</a:supportedlock>
<a:getcontenttype>text/plain</a:getcontenttype>
</a:prop>
</a:propstat>
</a:response>
</a:multistatus>
```

PROPFIND, named properties



PROPFIND /demo.txt HTTP/1.1

Host: dav.ics.uci.edu

Content-Type: text/xml; charset="utf-8"

Content-Length: 160

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

```
<d:propfind xmlns:d="DAV:">
```

```
  <d:prop>
```

```
    <d:getetag/>
```

```
    <d:getcontenttype/>
```

```
  </d:prop>
```

```
</d:propfind>
```

PROPFIND, named properties (2)



HTTP/1.1 207 Multi-Status

Server: DAV-demo-server/1.0

Date: Tue, 09 Feb 1999 00:52:55 GMT

Content-Type: text/xml; charset="utf-8"

Content-Length: 321

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

```
<a:multistatus xmlns:a="DAV:">
```

```
<a:response>
```

```
  <a:href>http://dav.ics.uci.edu/demo.txt</a:href>
```

```
  <a:propstat>
```

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

PROPFIND, named properties (3)



```
<a:prop>  
  <a:getetag>"10598798c153be1:a82"</a:getetag>  
  <a:getcontenttype>text/plain</a:getcontenttype>  
</a:prop>  
</a:propstat>  
</a:response>  
</a:multistatus>
```


PROPPATCH



PROPPATCH /webdav.html HTTP/1.1

Host: sandbox.xerox.com

Content-Type: text/xml; charset="utf-8"

Content-Length: 283

```
<?xml version="1.0" encoding="utf-8" ?>
<d:propertyupdate xmlns:d="DAV:">
  <d:set>
    <d:prop xmlns:j="http://www.ics.uci.edu/~ejw/">
      <j:personal>
        <j:item>My property</j:item>
      </j:personal>
    </d:prop>
  </d:set>
</d:propertyupdate>
```

PROPPATCH (2)



HTTP/1.1 207 Multi-Status

Date: Tue, 09 Feb 1999 01:36:43 GMT

Server: PyDAV 1.1 filestore 1.1

Content-Type: text/xml; charset="utf-8"

Content-Length: 317

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

```
<A:multistatus xmlns:A="DAV:">
```

```
  <A:response>
```

```
    <A:href>/webdav.html</A:href>
```

```
    <A:propstat>
```

```
      <A:prop>
```

```
        <B:personal xmlns:B="http://www.ics.uci.edu/~ejw/" />
```

```
      </A:prop>
```

PROPPATCH (3)



```
<A:status>HTTP/1.1 200 OK</A:status>  
</A:propstat>  
</A:response>  
</A:multistatus>
```



Collections

Collections



- **A collection is a set of URIs listing the member resources**
 - A resource may have more than one URI/URL, hence a resource may be accessible via more than one collection.
- **Advanced Collections specification defines ordered collections:**
 - The server persistently stores a single ordering of the resources in a collection.
 - A listing of the members of a collection returns the resources in that order.

Collection Benefits



- Provides a resource type which can model file system directories.
- Using referential resources, can create arbitrary collections of resources, which may be located on multiple servers.
- Provides hierarchical navigation capability
 - Supports “File... Save” or “File... Open” dialog box functionality in existing applications
- Provides scope for “Depth” operations

Collections: MKCOL



- **The MKCOL method creates a new collection**
- **MKCOL can be invoked with a request body**
 - **No request body: empty collection is created**
 - **Request body: media-type specific results**
 - primarily to reserve body for future capability
- **Benefits:**
 - a simple method is used to create a collection
 - MKCOL method avoids problems with adding collection creation semantics to PUT

Listing Members of a Collection: PROPFIND

- **PROPFIND lists the members of a collection, along with selected properties, when used with Depth 1 or Depth infinity**
 - **Depth 1: the collection and its members**
 - **Depth infinity: the collection and its members, and its children and their members, recursively**
- **Key insight: a directory listing function (ls, dir, etc.) is just a specialized metadata retrieval function**
- **Benefit: a tailorable collection listing operation**

Namespace Management: COPY



- **COPY method duplicates an HTTP resource, property or collection**
- **Recursive behavior for collections controlled by Depth header**
- **Copies are performed by-value**
- **Benefit:**
 - **allows duplication of resources without needing to transmit them over the wire twice.**

COPY Options



- **Overwrite header can disable overwriting the destination**
- **keepalive XML element (in request body) lists properties which must be live on the destination or the copy must fail**
- **Benefits:**
 - **prevent overwriting of existing data**
 - **fine-grain control over the copying of properties**

Namespace Management:

MOVE



- **MOVE is defined to be a COPY, followed by a DELETE, performed atomically**
- **All COPY options apply to MOVE as well**
- **Benefits:**
 - **allows the renaming of resources**
 - **allows positioning the resource in a different location in the hierarchy**
 - **more efficient than separate COPY and DELETE**



Overwrite Prevention: Locking

Write Lock

- **Write lock is the only lock access type defined in this specification**
- **A write lock prevents a principal without the lock from successfully modifying the state of the resource**
 - **Specifically, it prevents execution of PUT, POST, DELETE, MKCOL, PROPPATCH, MOVE, LOCK, UNLOCK**
 - **GET, PROPFIND are unaffected by write lock**

Write Lock Scope



- **The scope of a lock is an entire resource**
 - **It is not possible to specify a sub-resource lock.**

Why?

- **The ability to create a sub-resource lock requires knowledge about the content type being locked.**
- **WebDAV wanted to support all content types equally, not give preferential support to a few.**
- **Due to the large number of content types, support for sub-resource locks would have added a lot of extra semantics to the protocol. Some content types wouldn't have been supported.**

Write Lock Scope (2)



- **Further arguments against sub-resource locking:**
 - **Content types change rapidly (two revisions of HTML, XML during development of WebDAV protocol). Supporting one revision of a content type would make protocol brittle, quickly obsolete.**
 - **New content types would not be supported, yet new content types emerge constantly.**

Write Lock Scope (3)



■ Implications of whole-resource locking:

■ Pro:

- | Supports existing applications which operate on entire files, providing easy migration path to add WebDAV support
- | Handles all content types

■ Con:

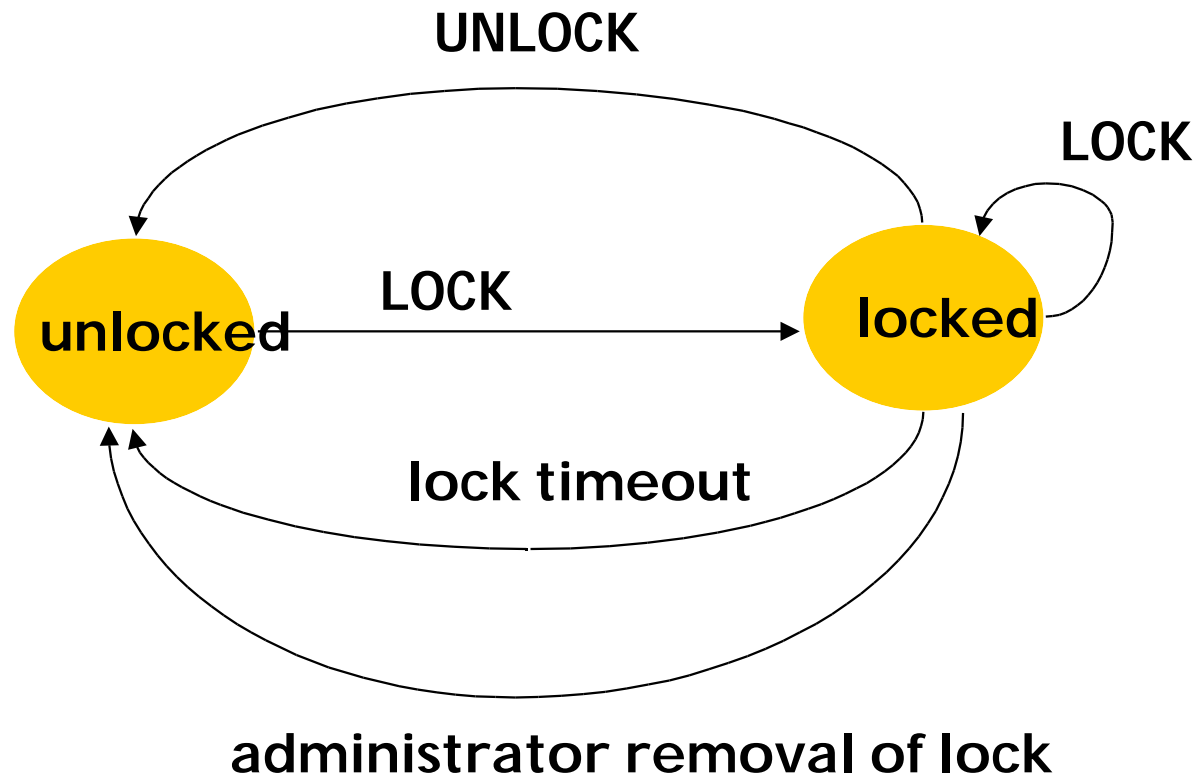
- | Reduced availability of resources during collaboration (but, shared locks can help...)

Write Lock (cont'd)



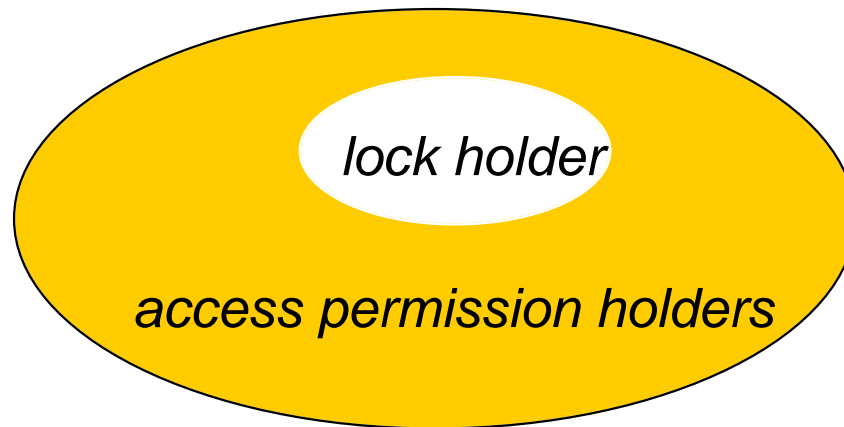
- **Live properties may still change even though a lock is active**
- **Dead properties may only be changed by the lock owner.**
- **A null resource may be locked to reserve its name. This makes the resource non-null, since it now has lock related properties defined on it.**

Lifecycle of a Lock



Exclusive Lock

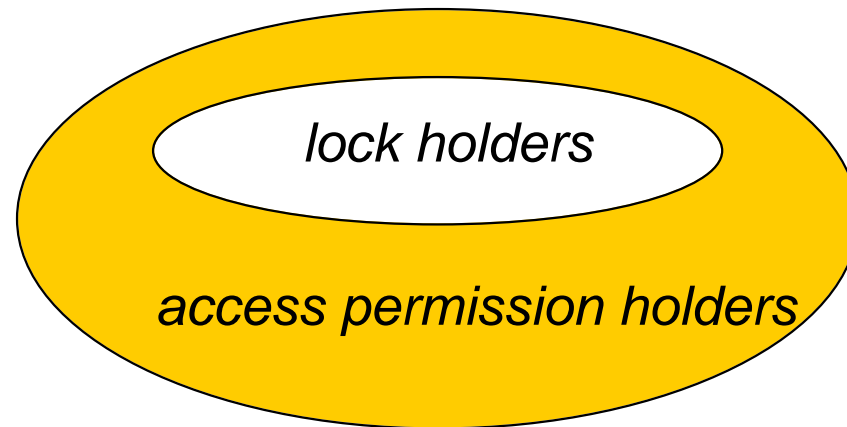
Zero or one possible lock holders



everyone on the Internet

Shared Lock

Zero, one, or many possible lock holders



everyone on the Internet

Why Exclusive and Shared?



- **Exclusive locks are too rigid**
 - people often forget to release locks
 - requires an administrator to release the lock
- **Shared locks allow people to use out-of-band communication to negotiate access to the resource**
 - if someone forgets to release a lock, it doesn't hold up the entire group
- **Collaborators work opportunistically**

Lock Compatibility

Current lock state \ Lock request	Shared Lock	Exclusive Lock
None	true	true
Shared Lock	true	false
Exclusive Lock	false	false*

Legend: True = lock MAY be granted
False = lock MUST NOT be granted

* = owner of lock MAY have lock regranted

LOCK Required Support



- A WEBDAV server is not required to support locking
- If a server does support locking, it may choose to support any combination of exclusive and shared locks.
- Why? Systems differ markedly in the type of locking support they provide, and may not be able to support locks at all (i.e., some replicated stores)

LOCK Method



- LOCK creates the lock specified by the `<lockinfo>` XML element (in the request body) on the Request-URI.
- A user-agent should submit owner information with a lock request
- LOCK returns a lock token which identifies the lock to the server
- The client may request a timeout value

Lock Owner Information



- The owner XML element (inside `<lockinfo>`) provides a means to associate lock holder contact information with a lock.
 - If you want the lock holder to release the lock, perhaps you can contact them and ask them to relinquish it
 - Authentication information often does not contain contact information (e.g., a key)

LOCK, single resource



LOCK /webdav.html HTTP/1.1

Host: sandbox.xerox.com

Timeout: Second-500, Infinite

Content-Type: text/xml

Content-Length: 151

```
<?xml version="1.0" ?>
```

```
<d:lockinfo xmlns:d="DAV:">
```

```
  <d:locktype><d:write/></d:locktype>
```

```
  <d:lockscope><d:exclusive/></d:lockscope>
```

```
</d:lockinfo>
```

LOCK, single resource (2)



HTTP/1.1 200 OK

Date: Tue, 09 Feb 1999 02:25:21 GMT

Server: PyDAV 1.1 filestore 1.1

Content-Type: text/xml

Content-Length: 435

```
<?xml version="1.0"?>  
<A:prop xmlns:A="DAV:">  
  <A:lockdiscovery>  
    <A:activelock>  
      <A:lockscope>  
        <A:exclusive/>  
      </A:lockscope>
```

LOCK, single resource (3)



```
<A:locktype>  
  <A:write/>  
</A:locktype>  
<A:depth>infinity</A:depth>  
<A:timeout>Second-500</A:timeout>  
<A:locktoken>  
  <A:href>opaquelocktoken:918527121.406</A:href>  
</A:locktoken>  
</A:activelock>  
</A:lockdiscovery>  
</A:prop>
```

Hierarchy Locks



- **Using the Depth header set to Infinity, can lock a collection hierarchy**
 - A single lock token is returned, identifying the lock on all resources.
 - An UNLOCK on this token removes the lock from all associated resources.
- **All or nothing semantics**
 - All resources in hierarchy are locked, or none are

Hierarchy Locks (cont'd)



- **Hierarchy locks act to ensure:**
 - All resources in the hierarchy are members of the lock
 - Resources removed from the hierarchy are removed from the lock
- **But...**
 - If a locked hierarchy is copied/moved, the destination hierarchy is not locked.

Hierarchy Lock Cases

■ Cases:

- **COPY/MOVE IN:** a resource copied/moved into a locked hierarchy is added to the lock for that hierarchy
- **COPY/MOVE WITHIN:** a resource copied/moved within a locked hierarchy is still a member of that hierarchy
- **COPY OUT:** when a resource is copied from a locked hierarchy, the source resource of the copy is still a member of the lock, the destination resource is not.

Hierarchy Lock Cases (cont'd)

- **MOVE OUT:** when a resource is moved from a locked hierarchy, it is removed from the source lock
- **DELETE:** removes the resource from the hierarchy lock
- **Combinations:**
 - Moving a resource from one locked hierarchy to another causes the resource to be removed from the source lock, and added to the destination lock.
 - Copying a resource from a locked hierarchy to another causes the source resource to remain in the source hierarchy lock, and the destination resource is added to the destination hierarchy lock.

UNLOCK



- **UNLOCK** removes the lock identified by a lock token from the Request-URI, and all other resources included in the lock
 - If a lock affects an entire collection, **UNLOCK** removes the lock from all resources in the collection.



Odds and ends...

Feature Discovery



- **HTTP OPTIONS method is applied to individual resources to discover supported capabilities**
 - **Allow header (HTTP/1.1) lists methods supported by the resource**
 - **DAV header (WebDAV) lists the WebDAV compliance level supported**
 - **1 - Meets all MUST requirements**
 - **2 - All MUSTs plus LOCK support**

OPTIONS Example



OPTIONS /wg/ HTTP/1.1

Host: www.webdav.org

HTTP/1.1 200 OK

Date: Tue, 09 Feb 1999 02:39:06 GMT

Server: Apache/1.3.4 (Unix) DAV/0.9.6

Content-Length: 0

**Allow: GET, HEAD, POST, PUT, DELETE, OPTIONS, TRACE, PROPFIND,
PROPPATCH, MKCOL, COPY, MOVE**

DAV: 1

Internationalization

- All WebDAV applications **MUST** support the character set tagging, character set encoding, and the language tagging functionality of the XML specification.
 - UTF-8 and UTF-16 charset support
 - UTF-16 begins with a BOM
 - Although the WebDAV spec. does not , it is strongly recommended to use the MIME charset parameter.
 - Examples:
 - `text/xml; charset="utf-8"`
 - `text/xml; charset="utf-16"`



Use Scenario: DAV Posties

Posties - Overview

■ Goal:

- **Create an electronic, collaborative Post-It™ note application**
 - Supports informal, lightweight interaction
- **The user should be able to just click on a note, and start typing**
 - But, the application should still prevent overwrites
- **Notes are shared by a group, which can be geographically dispersed**
 - One user's changes appear on others' screen
- **Synergy with Web - view notes in browser**

Posties: Why use DAV?



- **The Posties application uses DAV well**
 - **Format of notes is unconstrained**
 - | Requires use of HTTP PUT
 - | DAV locking prevents overwrites
 - **User interface metaphor doesn't map well to Web forms**
 - | Going to a page then changing a form too heavyweight, want to just click and edit
 - | Requires use of a custom user interface

Posties: Known Deficiencies of HTTP/DAV

- **HTTP/DAV doesn't provide notification of note updates**
 - HTTP/DAV is a request/response protocol, where the server cannot send asynchronous messages to the client
 - A separate notification protocol can provide this functionality, and there is much current work in this area (e.g. NSTP, GENA)
 - DAV does allow efficient polling for changes - use PROPFIND to request entity tags for all notes in one request

Posties: Use of DAV



- **Each note is a separate Web resource.**
 - Can be viewed by a Web browser, as well as by Posties application
 - All notes for a collaborative context are within a single collection.
- **When a user clicks on a resource, and starts typing, Posties checks for modifications to the note, uploading new note contents if necessary**

Posties: Use of DAV (2)

- **Overwrite prevention using locks**
 - After checking for new note contents, it requests a lock in the background (LOCK method)
 - If no lock exists, user is unaware lock took place
 - If lock exists, error dialog is displayed, small loss of typed data.
 - Lock is released when user clicks on other note, or is inactive (HTTP PUT, followed by UNLOCK)
- **Poll for changes to notes on a regular basis**
 - every 10 minutes, adjustable by user
 - Perform depth infinity PROPFIND for DAV:getetag
 - HTTP GET retrieves modified notes (locally store etag)

Posties: Other Issues



■ Privacy

- HTTP Digest authentication, in conjunction with server-side access control can be used to:
 - Protect privacy of notes
 - Limit set of people who can write to notes (delimit collaboration group)
 - Requires “log-in” step when first starting Posties
 - Need to integrate functionality from standard access control protocol, when its done



Summary

A Solid Specification



- **The base WebDAV specification is finished, and stable, the end product of significant review:**
 - Over 1,000 list messages
 - 15 meetings
 - Nine major revisions
 - Many independent reviewers
- **A solid specification**

Use your Imagination



- **Combine the power of the core Web for information distribution with...**
- **The ability to write content to a server under programmatic control.**
- **What do you get?**
 - **You decide!**

Getting Involved (WebDAV)

■ How do you join the WEBDAV Working Group?

■ Join the mailing list

■ `w3c-dist-auth@w3.org`

■ Send message with subject "subscribe" to `w3c-dist-auth-request@w3.org`

■ Attend a working group meeting:

■ Next meeting: Minneapolis, MN (USA) IETF, March 15-19, 1999

<http://www.ics.uci.edu/pub/ietf/webdav/>

Getting Involved (DASL)



- **How do you join the Distributed Authoring, Searching and Locating group?**
 - **Mailing list: www-webdav-dasl@w3.org**
 - **Send message with subject "subscribe" to www-webdav-dasl-request@w3.org**
- **Also meeting at Minneapolis IETF**
- **<http://www.ics.uci.edu/pub/ietf/dasl/>**

WebDAV Resources



- **WebDAV Working Group**
 - <http://www.ics.uci.edu/pub/ietf/webdav/>
- **webdav.org - WebDAV Resources**
 - <http://www.webdav.org/>
 - Maintained by Greg Stein
- **DASL Working Group**
 - <http://www.ics.uci.edu/pub/ietf/dasl/>
- **WebDAV Projects**
 - <http://www.webdav.org/projects/>
 - A complete list of known WebDAV projects, maintained by Michael Best

WebDAV Resources (2)



■ WebDAV Distributed Authoring Protocol

- <http://www.ics.uci.edu/pub/ietf/webdav/protocol/draft-ietf-webdav-protocol-10.pdf>
- <http://www.ics.uci.edu/pub/ietf/webdav/protocol/draft-ietf-webdav-protocol-10.txt>
- Soon to be released as RFC 2518:
 - <ftp://ftp.isi.edu/in-notes/rfc2518.txt>

■ Hypertext Transfer Protocol 1.1

- Proposed Standard Request for Comments 2068
- <ftp://ftp.isi.edu/in-notes/rfc2068.txt>
- <http://www.ics.uci.edu/pub/ietf/http/rfc2068.ps.gz>

WebDAV Resources (3)



- **XML 1.0 Recommendation**
 - <http://www.w3.org/TR/REC-xml/>
- **XML Namespaces Recommendation**
 - <http://www.w3.org/TR/REC-xml-names/>
- **XML Media Types**
 - Request for Comments (RFC) 2376
 - <ftp://ftp.isi.edu/in-notes/rfc2376.txt>

Mailing List Archives



■ WebDAV Mailing List

- w3c-dist-auth@w3.org

- <http://lists.w3.org/Archives/Public/w3c-dist-auth/>

■ DAV Versioning Mailing List

- ietf-dav-versioning@w3.org

- <http://lists.w3.org/Archives/Public/ietf-dav-versioning/>

■ DASL Mailing List

- www-webdav-dasl@w3.org

- <http://lists.w3.org/Archives/Public/www-webdav-dasl/>

WebDAV Introductory Articles



- **WebDAV: IETF Standard for Collaborative Authoring on the Web**
 - IEEE Internet, September/October, 1998
 - http://www.ics.uci.edu/pub/ietf/webdav/intro/webdav_intro.pdf
- **Collaborative Authoring on the Web: Introducing WebDAV**
 - ASIS Bulletin, Vol. 25, No. 1, Oct/Nov, 1998
 - <http://www.asis.org/Bulletin/Oct-98/webdav.html>