

INTERNET-DRAFT
[draft-ietf-clemm-versioning-01](#)
Expires September 22, 1999

Geoffrey Clemm
Rational Software
March 22, 1999

Versioning Extensions to WebDAV

Status of this Memo.

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

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

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

Abstract

This document specifies a set of methods, headers, and resource-types composing the DAV Versioning extensions to the HTTP/1.1 protocol. DAV Versioning will minimize the complexity of clients so as to facilitate widespread deployment of applications capable of utilizing the DAV Versioning services. DAV Versioning includes:

- Automatic versioning support for versioning-unaware clients,
- Multiple resource revisions with default revision selection,
- Resource branching for basic parallel development, and
- Configuration support for sophisticated parallel development.

Table of Contents

1	DAV VERSIONING PROTOCOL LEVELS.....	4
1.1	Version-Management Protocol.....	5
1.2	Configuration-Management Protocol.....	5
2	TERMS.....	5
2.1	Version-ManagementTerms.....	5
2.1.1	Versionable-Resource.....	5
2.1.2	Revision.....	5
2.1.3	Versioned-Resource.....	5
2.1.4	Versioning-Reference.....	5
2.1.5	Revision-Id.....	6
2.1.6	Label-Id.....	6
2.1.7	Working-Resource.....	6
2.1.8	Workspace.....	6
2.1.9	Default Workspace.....	6
2.1.10	Default Revision.....	7
2.2	Configuration-Management Terms.....	7
2.2.1	Activity.....	7
2.2.2	Repository.....	7
2.2.3	Versioned-Collection.....	7
2.2.4	Configuration.....	7
2.3	Notational Conventions.....	7
3	VERSIONING PROPERTIES.....	8
3.1	Property Attributes.....	8
3.1.1	Writeable/Readonly Properties.....	8
3.1.2	Immutable/Mutable Properties.....	8
3.1.3	Property-Collections.....	8
3.2	Versioned-Resource Properties.....	9
3.2.1	DAV:revisions (readonly, collection).....	9
3.2.2	DAV:revision-labels (collection).....	9
3.3	Revision Properties.....	9
3.3.1	DAV:versioned-resource-id (readonly, immutable)...	9
3.3.2	DAV:revision-id (readonly, immutable).....	9
3.3.3	DAV:predecessor (readonly, immutable).....	9
3.3.4	DAV:successors (readonly, mutable, collection)...	10
3.3.5	DAV:workspaces (readonly, mutable, collection)...	10
3.3.6	DAV:single-checkout.....	10
3.3.7	DAV:auto-version.....	10
3.4	Versioning-reference Properties.....	10
3.4.1	DAV:ref-target.....	10
3.4.2	DAV:default-workspace (readonly).....	10
3.5	Working-Resource Properties.....	10
3.5.1	DAV:workspace (readonly).....	11

3.5.2	DAV:predecessor (readonly).....	11
3.5.3	DAV:checkin-policy.....	11
3.6	Workspace Properties.....	11
3.6.1	DAV:ref-target (readonly).....	11
3.6.2	DAV:current.....	11
3.6.3	DAV:revision-selection-rule.....	11

4	VERSIONING METHODS	12
4.1	Existing Methods	12
4.1.1	GET	12
4.1.2	PUT	12
4.1.3	PROPPATCH	12
4.1.4	DELETE	12
4.1.5	COPY	13
4.1.6	MOVE	13
4.1.7	LOCK	13
4.2	New Versioning Methods	13
4.2.1	RESOURCEPUT	13
4.2.2	CHECKOUT	14
4.2.3	CHECKIN	14
4.2.4	UNCHECKOUT	14
4.3	New Versioning Headers	14
4.3.1	Version-Selector	14
5	CONFIGURATION-MANAGEMENT PROPERTIES	15
5.1	Resource Properties	15
5.1.1	DAV:workspaces (collection)	15
5.2	Versioned-resource Properties	15
5.2.1	DAV:versioned-resource-id (readonly)	15
5.2.2	DAV:configuration	15
5.3	Revision Properties	15
5.3.1	DAV:merge-predecessors (mutable, collection)	15
5.3.2	DAV:merge-successors (mutable, collection)	16
5.3.3	DAV:child-configurations (immutable, collection)	16
5.4	Working-Resource Properties	16
5.4.1	DAV:merge-predecessors (collection)	16
5.4.2	DAV:activity (readonly)	16
5.5	Workspace Properties	16
5.5.1	DAV:current	16
5.6	Activity Properties	17
5.6.1	DAV:revisions (readonly, collection)	17
5.6.2	DAV:child-activities (collection)	17
5.6.3	DAV:workspace	17
5.7	Repository Properties	17
5.7.1	DAV:versioned-resources (collection)	17
5.7.2	DAV:activities (collection)	17
6	CONFIGURATION-MANAGEMENT METHODS AND HEADERS	17
6.1	Existing Methods	17
6.1.1	RESOURCEPUT	18
6.1.2	PUT	18
6.1.3	COPY	18
6.1.4	MOVE	18
6.1.5	LOCK	18

6.1.6	CHECKOUT.....	18
6.1.7	CHECKIN.....	18
6.2	New Methods.....	19
6.2.1	GET-CONFLICTS.....	19
6.2.2	COMPARE.....	19
7	THE DAV VERSIONING XML ELEMENTS.....	19

7.1	Revision-Selection-Rule Elements.....	19
7.1.1	DAV:rsr-configuration-revision.....	19
7.1.2	DAV:rsr-activity.....	19
7.1.3	DAV:rsr-label.....	20
7.1.4	DAV:rsr-revision-id.....	20
7.1.5	DAV:rsr-latest.....	20
7.1.6	DAV:rsr-or-else.....	20
7.1.7	DAV:rsr-or-else-conflicts.....	20
7.2	Conflict Report Elements.....	21
7.2.1	DAV:conflict.....	21
7.2.2	DAV:contributor.....	21
7.2.3	DAV:contributor-base.....	21
7.3	Compare Report Elements.....	21
7.3.1	DAV:added.....	21
7.3.2	DAV:deleted.....	21
7.3.3	DAV:changed.....	21
8	INTERNATIONALIZATION CONSIDERATIONS.....	22
9	SECURITY CONSIDERATIONS.....	22
10	SCALABILITY.....	22
11	AUTHENTICATION.....	22
12	IANA CONSIDERATIONS.....	22
13	COPYRIGHT.....	22
14	INTELLECTUAL PROPERTY.....	22
15	REFERENCES.....	22
16	AUTHOR'S ADDRESS.....	23
17	OPEN ISSUES.....	23

[1](#) DAV VERSIONING PROTOCOL LEVELS

This document describes two classes of versioning support: version-management and configuration-management. To maximize interoperability and use of existing protocol functionality, the version-management protocol is an extension of the WebDAV advanced-collection protocol [[WEBDAV-ADVCOL](#)], and the configuration-management protocol is an extension of the version-management protocol. . These versioning protocols are designed so that WebDAV locking (class 2) support is optional. The effect of a lock on

versioning methods and content-types will be defined to provide interoperability of servers that provide locking support.

[1.1](#) Version-Management Protocol

Version-management is a basic versioning protocol designed to support versioning of largely independent resources. It allows authors to concurrently create and access distinct revisions of a resource.

[1.2](#) Configuration-Management Protocol

Configuration management is a more comprehensive versioning protocol, designed to support consistent versioning of sets of related resources being modified in parallel by many authors at different sites. Configuration management includes the capability of versioning the URL namespace to maintain the validity of both relative and absolute URL references.

[2](#) TERMS

This draft uses the terms defined in [\[RFC2068\]](#), [\[RFC2568\]](#), and [\[WEBDAV-ADVCOL\]](#). In addition, the following terms are introduced:

[2.1](#) Version-ManagementTerms

[2.1.1](#) Versionable-Resource

A versionable-resource is a resource that can be placed under version control. A null-resource is a versionable-resource.

[2.1.2](#) Revision

A revision is a resource that captures one state of a resource that is under version control. A revision is not a versionable-resource.

[2.1.3](#) Versioned-Resource

A versioned-resource is a type of resource that collects and controls all revisions of a resource that has been placed under version control. A versioned-resource is not a versionable-resource.

[2.1.4](#) Versioning-Reference

A versioning-reference is a special type of reference created when a versionable-resource is placed under version control. The DAV:ref-target of a versioning-reference is a versioned-resource but the target of a versioning-reference is a revision or working-resource of that versioned-resource. A Version-Selection header is used to specify the target of a versioning-reference. A versioning-reference is not a versionable-resource.

As with an ordinary reference, a DELETE or MOVE request is applied to the versioning-reference itself, while all other requests such as GET, PUT, PROPFIND, PROPPATCH, COPY, and LOCK are passed on to the target of the versioning-reference. This allows versioning-unaware clients to interact appropriately with a versioned-resource. In particular, it ensures that a versioning-unaware DELETE does not destroy the history captured by a versioned-resource.

[2.1.5](#) **Revision-Id**

A revision-id is a string specified by the server to identify a particular revision of a versioned-resource.

[2.1.6](#) **Label-Id**

A label-id is a string specified by the client to identify a particular revision of a versioned-resource.

[2.1.7](#) **Working-Resource**

A working-resource is a resource that is created by checking out a versioning-reference. A working-resource can be directly modified by a PUT request. A working-resource is not a versionable-resource.

[2.1.8](#) **Workspace**

A workspace is a type of resource that contains working-resources and is used to specify revision selection for versioning-references. The DAV:revision-selection-rule property of a workspace specifies how revision selection is done in that workspace. A workspace is not a versionable-resource.

[2.1.9](#) **Default Workspace**

A server MUST provide a default workspace that is used to perform version selection for versioning-unaware clients. The revision-selection-rule of the default workspace MAY be modifiable by a client. The DAV:current property of the default workspace must be set.

A workspace acts like a reference to the root collection of the web

site, except that for URL's that contain this reference, this workspace becomes the default workspace for versioning-unaware clients. As with an ordinary reference, a DELETE or MOVE request is applied to the workspace itself, while all other requests such as GET, PUT, PROPFIND, PROPPATCH, COPY, and LOCK are passed on to the root collection of the web site.

[2.1.10](#) **Default Revision**

The default revision of a versioning-reference is the one selected by the revision selection rule of the default workspace.

[2.2](#) **Configuration-Management Terms**

[2.2.1](#) **Activity**

An activity is a type of resource that is used to identify a set of revisions (possibly from multiple versioned-resources) that together define a consistent change. An activity is not a versionable-resource.

[2.2.2](#) **Repository**

A repository is a type of versioning-reference that controls a set of versioned-resources and activities. A repository is not a versionable-resource.

[2.2.3](#) **Versioned-Collection**

A versioned-collection is a type of versioned-resource created in a repository to collect and control revisions of a collection. An initial versioned-collection is created when a repository is created, and the DAV:ref-target of the repository is set to this versioned-collection. Each member of a versioned-collection revision either is a versioning-reference to a versioned-resource in that repository, or is a reference. Versioned-collections provide a mechanism for versioning a sub-tree of the URL namespace, and thereby maintain the validity of both relative and absolute URL references in distinct views of that URL namespace.

[2.2.4](#) **Configuration**

A configuration is a type of versioned-resource in a repository. A configuration revision is a collection that contains references to other revisions in that repository, where the name of each reference is the versioned-resource-id of the versioned-resource that contains that revision. This ensures that a configuration can contain at most one revision of a given versioned-resource, and provides a convenient protocol for determining the revision of a versioned-resource selected by the configuration.

[2.3](#) Notational Conventions

The augmented BNF used by this document to describe protocol elements is exactly the same as the one described in [Section 2.1 of \[RFC2068\]](#). Because this augmented BNF uses the basic production

rules provided in [Section 2.2 of \[RFC2068\]](#), those rules apply to this document as well.

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 [\[RFC2119\]](#).

[3](#) VERSIONING PROPERTIES

Version-management introduces the following additional properties:

[3.1](#) Property Attributes

There are several important attributes of properties that will be defined for every property introduced by this document.

[3.1.1](#) Writeable/Readonly Properties

A writeable property can be modified by a client, while a readonly property can only be modified by the server. All properties defined in this document are writeable unless explicitly indicated as readonly.

[3.1.2](#) Immutable/Mutable Properties

An immutable resource is a resource whose value cannot change. An immutable property is a property whose value cannot change when it appears on an immutable resource. A mutable property is a property whose value can change, even when it appears on an immutable resource. All properties defined in this document are immutable unless explicitly indicated as mutable.

[3.1.3](#) Property-Collections

The value of a property can be the URL of a collection, where this URL value cannot be modified by a client. Such a property is called a property-collection. The writeable/readonly and immutable/mutable attributes of a property-collection indicate whether members can be added or removed from the property-collection. All properties defined in this document are not property-collections, unless explicitly indicated to be one.

When the members of a property-collection are all references, then the property-collection could be represented as a comma-separated

list of URL's or as an XML document, but this is susceptible to lost updates (when one client updates the value without locking the property before reading it). A comma-separated list of URL s also cannot handle the case where an immediate member is a resource and not just a reference to a resource.

3.2 Versioned-Resource Properties

Version-management introduces the following additional properties for a versioned-resource:

3.2.1 DAV:revisions (readonly, collection)

The DAV:revisions collection of a versioned-resource contains all revisions of that versioned-resource, where the name of a revision in the DAV:revisions collection is its DAV:revision-id. PROPFIND can then be used to gather arbitrary information about the revisions. If a revision-id contains a URL reserved character, that character is escaped in the DAV:revisions name.

3.2.2 DAV:revision-labels (collection)

The DAV:revision-labels collection of a versioned-resource contains references to members of the DAV:revisions collection of that versioned-resource. The client can label and unlabel revisions by adding and deleting members of the DAV:revision-labels collection. If a label-id contains a URL reserved character, that character is escaped in the DAV:revision-labels name.

3.3 Revision Properties

Version-management introduces the following additional properties for a revision:

3.3.1 DAV:versioned-resource-id (readonly, immutable)

The DAV:versioned-resource-id of a revision is the DAV:versioned-resource-id of the versioned-resource whose DAV:revisions collection contains this revision.

3.3.2 DAV:revision-id (readonly, immutable)

The DAV:revision-id is an identifier assigned to a revision by the server. Whenever a revision is created or modified by a CHECKIN, it must be assigned a new DAV:revision-id. A revision cannot be given a DAV:revision-id that has been given to any other revision of that versioned-resource, even a revision that has been deleted.

3.3.3 DAV:predecessor (readonly, immutable)

The DAV:predecessor of a revision is the revision that was checked out to produce a working resource that was checked in to produce this revision.

[3.3.4](#) **DAV:successors (readonly, mutable, collection)**

The DAV:successors collection of a revision contains a reference to each revision whose DAV:predecessor is that revision.

[3.3.5](#) **DAV:workspaces (readonly, mutable, collection)**

The DAV:workspaces collection contains references to all workspaces that have a working-resource checked out from this revision.

[3.3.6](#) **DAV:single-checkout**

When the DAV:single-checkout property of a revision is set, only one working-resource can be checked out from that revision at any time.

[3.3.7](#) **DAV:auto-version**

When the DAV:auto-version property of a revision is set, a PUT request to this revision is automatically preceded by a CHECKOUT into the default workspace and followed by a CHECKIN. This allows a versioning-unaware client to modify a version-controlled resource. The DAV:auto-version value can take the same values as the DAV:checkin-policy of a working-resource, and the DAV:checkin-policy of the automatically created working-resource is set to be the DAV:auto-version policy of the revision.

[3.4](#) **Versioning-reference Properties**

Version-management introduces the following additional properties for a versioning-reference:

[3.4.1](#) **DAV:ref-target**

The DAV:ref-target of a versioning-reference is the versioned-resource that contains the revisions that may be the target of this versioning-reference.

[3.4.2](#) **DAV:default-workspace (readonly)**

The DAV:default-workspace property specifies which workspace will be used to perform version selection for a versioning-reference request that does not contain an explicit Version-Selector header.

3.5 Working-Resource Properties

Version-management introduces the following additional properties for a working-resource:

[3.5.1](#) **DAV:workspace (readonly)**

The DAV:workspace of a working-resource is the workspace that contains this working resource.

[3.5.2](#) **DAV:predecessor (readonly)**

The DAV:predecessor of a working-resource is the revision that was checked-out to produce that working-resource.

[3.5.3](#) **DAV:checkin-policy**

The DAV:checkin-policy property of a working-resource indicates how this working-resource should be checked in. The default value is DAV:immutable.

DAV:identical-abort indicates that the CHECKIN should fail if the working-resource is identical to its DAV:predecessor.

DAV:identical-uncheckout indicates that an UNCHECKOUT should be applied if the working-resource is identical to its DAV:predecessor.

DAV:overwrite indicates that the working resource should be copied into the DAV:predecessor instead of creating a new revision.

DAV:mutable indicates that the new revision can be overwritten by a subsequent DAV:overwrite CHECKIN.

[3.6](#) **Workspace Properties**

Version-management introduces the following additional properties for a workspace:

[3.6.1](#) **DAV:ref-target (readonly)**

The DAV:ref-target of a workspace is the root collection of the web site (i.e. "/"). This reflects the fact that a workspace can be used to specify version selection for any versioning-reference at a web site. Since a workspace is a reference, the properties of a workspace must be accessed with a DAV:nopassthrough header.

[3.6.2](#) **DAV:current**

The DAV:current property of a workspace can contain the name of a label. Whenever a working-resource is checked in to create a new

revision, the DAV:current label of the workspace is applied to that revision.

3.6.3 DAV:revision-selection-rule

The DAV:revision-selection-rule of a workspace contains an XML document that describes how revision selection will be performed in that workspace. The working-resources checked-out into a workspace

take priority over revisions selected by the revision-selection-rule, thus the version selection algorithm for a workspace is "checked-out to that workspace" else "selected by the workspace revision-selection rule". To ensure that working-resources continue to be visible in a workspace after they are checked in, the contents of DAV:current is usually the first element of the DAV:revision-selection-rule. If the DAV:revision-selection-rule is not set or is empty, the revision selection rule will select no revision for any versioning-reference.

Standard revision-selection-rule elements are defined in this document, but additional revision-selection-rule elements may be supported by a WebDAV server.

4 VERSIONING METHODS

4.1 Existing Methods

4.1.1 GET

When GET is applied to a versioning-reference, it returns the body of the target of that versioning-reference.

4.1.2 PUT

When PUT is applied to a versioning-reference whose target is a working-resource, the PUT is applied to that working-resource. When PUT is applied to a versioning-reference whose target is a revision, the PUT fails unless the revision has a DAV:auto-version property and no Version-Selector header has been specified. In this case, the versioning-reference is checked out into the default workspace, the PUT is applied to the resulting working-resource, and the working-resource is checked in. When PUT is applied to a versioned-resource or a revision, it fails.

4.1.3 PROPPATCH

When PROPPATCH is applied to a versioning-reference, its behavior is similar to that of PUT. When PROPPATCH is applied to a revision, it fails unless the property is mutable.

4.1.4 DELETE

When DELETE is applied to a working-resource or a versioning-reference whose target is a working-resource, the workspace of the working-resource is deleted from the DAV:workspaces of the DAV:predecessor of the working-resource and the working-resource is deleted. When DELETE is applied to a versioning-reference whose target is a revision, the versioning-reference is deleted, but the revision is unaffected. When DELETE is applied to a workspace, the

workspace and all working-resources of that workspace are deleted. When DELETE is applied to a revision, the revision is deleted. When DELETE is applied to a versioned-resource, the versioned-resource and all revisions of that versioned-resource are deleted.

[4.1.5](#) **COPY**

When COPY is applied to a versioned-resource, it fails.

[4.1.6](#) **MOVE**

When MOVE is applied to a versioning-reference, the versioning-reference is moved, but the versioned-resource and revisions of that versioning-reference are unaffected. When MOVE is applied to a working-resource, the working-resource is moved but its associated versioned-resource is unaffected. When a MOVE is applied to a versioned-resource, workspace, or revision, it fails.

[4.1.7](#) **LOCK**

A write-lock on a versioned-resource places a write-lock on all revisions of that versioned-resource. A write-lock on a revision prevents unauthorized modifications to the DAV:workspaces property of that revision. A write-lock on a working-resource prevents unauthorized changes to the body or properties of that working-resource.

[4.2](#) **New Versioning Methods**

Version-management introduces the following additional methods:

[4.2.1](#) **RESOURCEPUT**

RESOURCEPUT is a generalized PUT that allows the update or creation of an arbitrary resource type. The body of RESOURCEPUT has the same syntax as PROPPATCH except that a DAV:body element can be used to update or initialize the body of the resource. If RESOURCEPUT is applied to a null resource, a new resource is created. If it is applied to a resource of the type specified by the RESOURCEPUT, the existing resource is updated.

If RESOURCEPUT is used to create a new versioning-reference, and the DAV:versioned-resource is not specified, then the result of the

RESOURCEPUT depends on whether the specified URL is a null resource, a versionable resource, or a versioning-reference. If it is a null resource, a new versioned-resource is created whose initial revision has a null body. If it is a versionable-resource, a new versioned-resource is created whose initial revision is a copy of that versionable-resource. If it is a versioning-reference, the versioning-reference is updated by the RESOURCEPUT.

[4.2.2](#) **CHECKOUT**

When a CHECKOUT request is applied to a versioning-reference whose target is a revision, a new working-resource is created that is a copy of the revision, and the DAV:predecessor of the working-resource is set to be that revision. If the DAV:predecessor has a DAV:single-checkout property and is already checked out into a workspace, the CHECKOUT request fails. The DAV:workspace of the working-resource is set to be the workspace specified in the Version-Selector header. If the Version-Selector is not a workspace or if there is no Version-Selector header, the DAV:workspace for that working-resource is allocated by the server. The body of the CHECKOUT request can be used to initialize the DAV:checkin-policy of the working-resource. The location of the new working-resource is returned in the Location response header. A client may indicate a preference for the location of the new working-resource with a Location request header, but the server MAY ignore this preference.

When a CHECKOUT request is applied to a versioning reference whose target is a working-resource, the CHECKOUT request fails.

[4.2.3](#) **CHECKIN**

When a CHECKIN request is applied to a working-resource of a versioned-resource, a copy of the working-resource is made a new revision of that versioned-resource and the working-resource is deleted. A reference to the new revision is added to the DAV:successors collection of the DAV:predecessor of the new revision, and the workspace of the working-resource is deleted from the DAV:workspaces collection of the DAV:predecessor.

[4.2.4](#) **UNCHECKOUT**

When an UNCHECKOUT request is applied to a working-resource or a versioning-reference whose target is a working-resource, the DAV:workspace of that working-resource is deleted from the DAV:workspaces collection of the DAV:revision of the working-resource, and the working-resource is deleted. This effectively cancels the CHECKOUT request that produced the working-resource.

[4.3](#) **New Versioning Headers**

[4.3.1](#) **Version-Selector**

The Version-Selector header contains a workspace URL, a label, or a revision-identifier. The Version-Selector header is used to specify which working-resource or revision of a versioned-resource should be the target of a versioning-reference. If a Version-Selector header is omitted in a request on a versioning-reference, the default-workspace is implicitly used to perform version selection.

5 CONFIGURATION-MANAGEMENT PROPERTIES

Configuration-management introduces the following additional properties:

5.1 Resource Properties

Configuration-management introduces the following additional properties for a resource:

5.1.1 DAV:workspaces (collection)

The DAV:workspaces collection of a resource contains workspaces.

5.2 Versioned-resource Properties

Configuration-management introduces the following additional properties for a versioned-resource:

5.2.1 DAV:versioned-resource-id (readonly)

The DAV:versioned-resource-id is an identifier assigned to a versioned-resource in a repository by the server. A versioned-resource cannot be given a DAV:versioned-resource-id that has been given to any other versioned-resource in that repository, even a versioned-resource that has been deleted.

5.2.2 DAV:configuration

The DAV:configuration of a versioned-collection is a configuration. When a DAV:configuration of a versioned-collection is checked out into a workspace, it tracks the revision selected by the workspace for each member of that collection. When the configuration is checked in, the new configuration revision can be used in a DAV:revision-selection-rule to select the revisions that were selected by the workspace at the time that configuration revision was checked in.

5.3 Revision Properties

Version-management introduces the following additional properties for a revision:

5.3.1 DAV:merge-predecessors (mutable, collection)

The DAV:merge-predecessors collection of a revision contains references to the other revisions whose contents were explicitly merged by the client into that revision. The client is free to add or delete members to this collection to more accurately reflect the contents of that revision.

[5.3.2](#) **DAV:merge-successors (mutable, collection)**

The DAV:merge-successors collection of a revision contains a reference to each revision whose DAV:merge-predecessors collection contains a reference to that revision.

[5.3.3](#) **DAV:child-configurations (immutable, collection)**

The DAV:child-configurations collection of a configuration contains references to revisions of configurations in other repositories. The DAV:child-configurations of a configuration revision contribute to the revision selection behavior of that configuration revision when it is used in a revision-selection-rule.

[5.4](#) **Working-Resource Properties**

Configuration-management introduces the following additional properties for a working-resource:

[5.4.1](#) **DAV:merge-predecessors (collection)**

The DAV:merge-predecessors collection of a working-resource contains references to the revisions whose contents were explicitly merged by the client into that working-resource. The client is free to add or delete members to this collection to more accurately reflect the contents of that working-resource. The name of a DAV:merge-predecessors member is the DAV:revision-id of that revision.

[5.4.2](#) **DAV:activity (readonly)**

The DAV:activity property of a working-resource contains the DAV:current activity of the workspace when the working-resource was checked out.

[5.5](#) **Workspace Properties**

Configuration-management introduces the following additional properties for a workspace:

[5.5.1](#) **DAV:current**

The DAV:current property of a workspace on a configuration-

management server can contain the URL of an activity. When DAV:current is an activity, every new working-resource in a workspace will have its DAV:activity property set to this activity.

5.6 Activity Properties

Configuration-management introduces the following additional properties for an activity:

5.6.1 DAV:revisions (readonly, collection)

The DAV:revisions collection of an activity contains references to all revisions whose DAV:activity property contains this activity.

5.6.2 DAV:child-activities (collection)

The DAV:child-activities collection of an activity contains references to other activities that form a part of the logical change being captured by the activity. The DAV:child-activities of an activity contribute to the revision selection behavior of that activity when it is used in a revision-selection-rule.

5.6.3 DAV:workspace

The DAV:workspace property of an activity contains the workspace that currently has that activity as its DAV:current activity. This implies that at most one workspace can be working in a given activity at a time. If any working-resource of a workspace is checked-out into a given activity, the DAV:workspace of that activity can only be that workspace.

5.7 Repository Properties

5.7.1 DAV:versioned-resources (collection)

The DAV:versioned-resources collection of repository contains the versioned-resources created in that repository.

5.7.2 DAV:activities (collection)

The DAV:activities collection of a repository contains the activities created in that repository.

6 CONFIGURATION-MANAGEMENT METHODS AND HEADERS

6.1 Existing Methods

When invoked on a configuration-management server, the following methods will display additional behavior.

[6.1.1](#) **RESOURCEPUT**

RESOURCEPUT can be used to create a new activity in a repository DAV:activities collection. When RESOURCEPUT is used to create a repository from an existing versionable collection, every member of that versionable collection is also placed under version control as a versioned-resource in that repository.

[6.1.2](#) **PUT**

When a PUT request is applied to an activity, it fails. When a PUT request is applied to a null-resource in a versioned-collection revision, the PUT request fails unless DAV:auto-version is set on that revision. If DAV:auto-version is set, a new versioned-resource is created in the repository, an initial revision whose body contains the body of the PUT request is created for that versioned-resource, and a versioning-reference to the new versioned-resource is created at the URL of the PUT.

[6.1.3](#) **COPY**

When a COPY request is applied to an activity or a repository, it fails.

[6.1.4](#) **MOVE**

When a MOVE request is applied to an activity, it fails.

[6.1.5](#) **LOCK**

A write lock on an activity prevents a modification to the DAV:revisions collection of that activity.

[6.1.6](#) **CHECKOUT**

On CHECKOUT, the DAV:activity of the new working-resource is set to be the DAV:current activity of the current workspace. If DAV:current is a label, the server automatically assigns an activity to the new working-resource. The CHECKOUT request fails if the workspace DAV:current is not set.

[6.1.7](#) **CHECKIN**

When a CHECKIN request is invoked on a working-resource of a versioned-collection, every versionable-resource in that collection is put under version control. When a CHECKIN request is invoked on a working-resource for a configuration of a versioned-collection, the new configuration revision selects the revisions currently selected by the workspace for that versioned-collection and each of its members.

[6.2](#) New Methods

Configuration-management introduces the following additional methods:

[6.2.1](#) GET-CONFLICTS

A GET-CONFLICTS request must have a Version-Selector header specifying a workspace. The body of the GET-CONFLICTS response contains an XML document describing all revision selection conflicts in that workspace. This document enumerates the versioning-references which have conflicts, as well as information about how to resolve the conflicts (such as the revisions that would need to be merged). If the body of a GET-CONFLICTS request is non-empty, it contains a revision selection rule that is appended by a DAV:rsr-or-else-conflicts to the DAV:revision-selection-rule of the workspace.

[6.2.2](#) COMPARE

A COMPARE request can be applied to two configuration revisions. The COMPARE response body contains an XML document listing the revisions, child-configuration-revisions, and activities that are selected by one configuration revision but not the other.

[7](#) THE DAV VERSIONING XML ELEMENTS

[7.1](#) Revision-Selection-Rule Elements

A revision-selection-rule (or RSR) element can be applied to an arbitrary versioning-reference, and will either select a particular revision of the versioned-resource of that versioning-reference or select no revision of that versioned-resource. If it selects a particular revision, it may also detect a conflict (e.g. when there were multiple candidates for selection). A document describing all the conflicts can be obtained through the GET-CONFLICTS method.

[7.1.1](#) DAV:rsr-configuration-revision

A DAV:rsr-configuration element contains the URL of a configuration revision. If the configuration revision contains a revision of the versioned-resource, that revision is selected by the DAV:rsr-configuration-revision element; otherwise, no revision is selected.

[7.1.2](#)

DAV:rsr-activity

A DAV:rsr-activity element contains the URL of an activity. If the DAV:revisions collection of the activity contains one or more revisions of the versioned-resource, then the revision is selected that is the descendent of each of the other revisions in the

DAV:revisions collection. The semantics of activities ensures that such a revision always exists in the DAV:revisions collection of an activity. If the activity contains no revisions of a versioned-resource, the DAV:rsr-activity element selects no revisions of that versioned-resource.

If the DAV:child-activities collection of an activity is non-empty, then the DAV:rsr-activity element acts like a DAV:rsr-or-else-conflicts element that contains that activity and each of the DAV:child-activities. A DAV:rsr-activity element can generate conflicts only if the DAV:child-activities collection is non-empty.

[7.1.3](#) **DAV:rsr-label**

A DAV:rsr-label element contains a label. If a revision of the versioned-resource has that label, that revision is selected by the DAV:rsr-label element; otherwise, no revision is selected. A DAV:rsr-label element never generates a conflict.

[7.1.4](#) **DAV:rsr-revision-id**

A DAV:rsr-revision-id element contains a revision-id. If a revision of the versioned-resource has that revision-id, that revision is selected by the DAV:rsr-revision-id element; otherwise, no revision is selected. A DAV:rsr-revision-id element never generates a conflict.

[7.1.5](#) **DAV:rsr-latest**

A DAV:rsr-latest element selects the revision of that versioned-resource with the latest DAV:modification-time.

[7.1.6](#) **DAV:rsr-or-else**

A DAV:rsr-or element contains a sequence of RSR elements. The behavior of a DAV:rsr-or element is identical to the behavior of the first element in this sequence that selects a revision of the versioned-resource. If no element selects a revision, then the DAV:rsr-or element selects no revision of that versioned-resource.

[7.1.7](#) **DAV:rsr-or-else-conflicts**

A DAV:rsr-or-else-conflicts element contains a sequence of RSR

elements. If one of the elements in this sequence selects a revision that is the descendent of all of the other revisions selected by elements in this sequence, then that revision is selected by the DAV:rsr-or-else-conflicts element. If no selected revision is a descendent of all the other selected revisions, then a conflict has occurred, and the DAV:rsr-or-else-conflicts selects the first revision selected by an element in the sequence. The GET-CONFLICTS method can be used to identify the conflicts.

7.2 Conflict Report Elements

A conflict report identifies the versioned-resources in a workspace that generate a conflict when the specified revision-selection-rule is applied to them.

[7.2.1](#) **DAV:conflict**

The DAV:conflict element identifies a particular versioned-resource for which the revision-selection-rule generated a conflict.

[7.2.2](#) **DAV:contributor**

The DAV:contributor element identifies a revision that is selected by an element of the revision-selection-rule but that is not an ancestor of any of the other revisions selected by the revision-selection-rule.

[7.2.3](#) **DAV:contributor-base**

The DAV:contributor-base element identifies a revision that is a common ancestor of each DAV:contributor for a particular conflict..

7.3 Compare Report Elements

A compare report identifies the differences between two resources of the same type.

[7.3.1](#) **DAV:added**

A DAV:added element identifies something that appears in the second resource but not in the first.

[7.3.2](#) **DAV:deleted**

A DAV:deleted element identifies something that appears in the first resource but not in the second.

[7.3.3](#) **DAV:changed**

A DAV:changed element identifies something that is in both resources but that is in some way different in the two resources.

For example, when two configurations are being compared, a DAV:changed element can identify a versioned-resource, a versioned-collection, or an activity. A versioned-resource has changed if the configurations select different revisions of that versioned-resource. A versioned-collection has changed if the configurations select different revisions or different configurations of that versioned-collection. An activity has changed if both configurations contained that activity but the DAV:revisions or

DAV:child-activities of that activity were different when those configurations were created.

8 INTERNATIONALIZATION CONSIDERATIONS

To be supplied.

9 SECURITY CONSIDERATIONS

To be supplied.

10 SCALABILITY

To be supplied.

11 AUTHENTICATION

Authentication mechanisms defined in WebDAV will also apply to DAV Versioning.

12 IANA CONSIDERATIONS

This document uses the namespace defined by [WebDAV] for XML elements. All other IANA considerations mentioned in [WebDAV] also applicable to DAV Versioning.

13 COPYRIGHT

To be supplied.

14 INTELLECTUAL PROPERTY

To be supplied.

15 REFERENCES

[RFC2068] 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.

[RFC2119] S. Bradner, "Key words for use in RFCs to Indicate

Requirement Levels." [RFC 2119](#), [BCP 14](#). Harvard University. March, 1997.

[RFC2518] Y. Goland, E.J. Whitehead, A. Faizi, S.R. Carter, D. Jenson, "HTTP Extensions for Distributed Authoring - WEBDAV", [RFC 2518](#), Microsoft, UC Irvine, Netscape, Novell, February. 1999.

[WEBDAV-ADVCOL]

16 AUTHOR'S ADDRESS

Geoffrey Clemm
Rational Software
20 Maguire Road
Lexington, MA 02421
Email: geoffrey.clemm@rational.com

