[MS-XWDREPL]:
Web Distributed Authoring and Versioning (WebDAV) Extensions for Replication

---

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
12/03/2008	1.0		Initial Release.
03/04/2009	1.01		Revised and edited technical content.
04/10/2009	2.0		Deprecated for Exchange 2010.
07/15/2009	3.0	Major	Changes made for template compliance.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	5.0.0	Major	Updated and revised the technical content.
05/05/2010	5.1.0	Minor	Updated the technical content.
08/04/2010	5.1.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/03/2010	5.1.0	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	5.2	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction

The Web Distributed Authoring and Versioning (WebDAV) Extensions for Replication extend the **HTTP** protocol and the **Web Distributed Authoring and Versioning Protocol (WebDAV)** to allow client-server replication of Web **resources** on a **WebDAV server**.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Augmented Backus-Naur Form (ABNF)
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
resource
Secure Sockets Layer (SSL)
universally unique identifier (UUID)

The following terms are defined in [MS-OXGLOS]:

binary large object (BLOB)
Transport Layer Security (TLS)
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
Web Distributed Authoring and Versioning Protocol (WebDAV)
WebDAV client
WebDAV server

The following terms are specific to this document:

optimistic concurrency: A model for updating data in a database that does not lock records and allows for improved performance.

paged results: A data model that allows a client to request that a server return a subset of a result set instead of the entire result set.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-WDVSE] Microsoft Corporation, "Web Distributed Authoring and Versioning (WebDAV) Protocol: Server Extensions", September 2007.

[MS-XWDMAIL] Microsoft Corporation, "Web Distributed Authoring and Versioning (WebDAV) Extensions for E-Mail Support", December 2008.

[MS-XWDSEARCH] Microsoft Corporation, "Web Distributed Authoring and Versioning (WebDAV) Extensions for Search", December 2008.

[RFC2068] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997, http://www.ietf.org/rfc/rfc2068.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC2246] Dierks, T., and Allen, C., "The TLS Protocol Version 1.0", RFC 2246, January 1999, http://www.ietf.org/rfc/rfc2246.txt

[RFC2518] Goland, Y., Whitehead, E., Faizi, A., et al., "HTTP Extensions for Distributed Authoring - WebDAV", RFC 2518, February 1999, http://www.ietf.org/rfc/rfc2518.txt

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, http://www.ietf.org/rfc/rfc5234.txt

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", April 2008.

[MS-XWDEXT] Microsoft Corporation, "Web Distributed Authoring and Versioning (WebDAV) Core Extensions", July 2009.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.ietf.org/rfc/rfc2818.txt

1.3 Overview

The Web Distributed Authoring and Versioning (WebDAV) Extensions for Replication are a set of methods, headers, and elements that extend the Hypertext Transport Protocol – HTTP/1.1, as described in [RFC2068]. WebDAV allows for the writing of data to Internet servers.

WebDAV replication is applied over the existing WebDAV operations that allow clients to do the following:

- Determine what has changed in a given collection, as described in [RFC2518].
- Update items by using **optimistic concurrency**.
- Locate and resolve conflicted items.

1.4 Relationship to Other Protocols

The WebDAV Extensions for Replication rely on WebDAV, as described in [RFC2518], which in turn relies on HTTP 1.1, as described in [RFC2068]. These extensions can use **HTTPS** for data protection, as described in [RFC2818].

The WebDAV Extensions for Replication are also dependent on the client, server, and Microsoft extensions to [RFC2518], as described in [MS-XWDEXT].

1.5 Prerequisites/Preconditions

The WebDAV Extensions for Replication require a WebDAV server, as described in [RFC2518]. These extensions also require that **WebDAV clients** have **URLs** that point to WebDAV servers.

1.6 Applicability Statement

The WebDAV Extensions for Replication are applicable in scenarios that require client applications to synchronize data on a WebDAV server.

1.7 Versioning and Capability Negotiation

Clients can determine whether a server supports the replication extensions by sending an **OPTIONS** command, as described in [RFC2068], to the server and examining the response. For a server to declare that it implements replication, it has to return "http://schemas.microsoft.com/repl/" in the **Public-Extension** header. If the server supports the WebDAV Extensions for Replication, it has to return an **Allow-Extension** header with the value "http://schemas.microsoft.com/repl/" in it. For information about these headers, see [MS-XWDMAIL] section 2.2.1.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Messages are transported by using HTTP, as specified in [RFC2518] and [RFC2068].

The WebDAV Extensions for Replication can be used with **Secure Sockets Layer (SSL)** or **Transport Layer Security (TLS)**, as specified in [RFC2246].

Port 80 is the standard port assignment for HTTP, and port 443 is the standard port assignment for HTTP over SSL or TLS; however, individual implementations MAY support other ports.

2.2 Message Syntax

The extension headers in the WebDAV Extensions for Replication conform to the form and behavior of other custom HTTP headers, as specified in [RFC2068] section 4.2, and are consistent with the WebDAV verbs and headers, as specified in [RFC2518] sections 8 and 9.

2.2.1 Headers

The **Augmented Backus-Naur Form (ABNF)** notation, as specified in [RFC5234], is used to specify the format of the **Range** header specified in section 2.2.1.1.

2.2.1.1 Range

A WebDAV replication–compliant server MUST implement the **paged results** and the **Range** header, as specified in [MS-XWDSEARCH], in order to improve the scalability and performance of the server. Clients SHOULD use the **Range** header and paged results to reduce the load on the server.

2.2.2 XML Elements

2.2.2.1 changetype

Namespace: http://schemas.microsoft.com/repl/

Syntax: <!ELEMENT changetype (delete | change | new | read) >

The purpose of the **changetype** XML element is for the server to indicate to the client the type of the change on a resource when the client retrieves the manifest of a collection.

2.2.2.2 collblob

Namespace: http://schemas.microsoft.com/repl/

Syntax: <!ELEMENT collblob (EMPTY | (#PCDATA)) >

The purpose of the **collblob** XML element is for the client to indicate that it wants to fetch a manifest from the server. The value of the **collblob** XML element is used by the client to provide its original replication state and by the server to indicate the client's updated replication state.

2.2.2.3 contenttag

Namespace: http://schemas.microsoft.com/repl/

Syntax: <!ELEMENT contenttag (#PCDATA) >

A **contenttag** XML element contains a value generated by the server to represent the state of the contents of a WebDAV collection, as specified in [RFC2518]. This XML element is applied only to resources immediately subordinate to the target **URI**, but the target resource itself is excluded.

2.2.2.4 repl

Namespace: http://schemas.microsoft.com/repl/

Syntax: <!ELEMENT repl (changetype | collblob | resourcetaglist)>

The **repl** element specifies the replication properties to be returned from a **SEARCH** method, as specified in [RFC2518].

2.2.2.5 repl-uid

Namespace: http://schemas.microsoft.com/repl/

Syntax: <!ELEMENT repl-uid (#PCDATA)>

The **repl-uid** element is a **universally unique identifier (UUID)** of a resource. The value of this property is a URI.

2.2.2.6 resourcetag

Namespace: http://schemas.microsoft.com/repl/

Syntax: <!ELEMENT resourcetag (#PCDATA) >

A **resourcetag** XML element contains a value generated by the server to represent the state of a WebDAV resource. This XML element is applied only to the resource. The value of this element is a URI.

The client SHOULD keep this value to reflect the state of the replicated resource. The following is a list of functions that the **resourcetag** XML element serves:

- A WebDAV client that wants to avail itself of the server-side conflict detection and resolution mechanism SHOULD send its previously obtained **resourcetag** XML element value in the request headers of the following requests: **GET** method, as specified in section 2.2.3.2; **PUT** method, as specified in section 2.2.3.8; **POST** method, as specified in section 2.2.3.5; **PROPFIND** method, as specified in section 2.2.3.6; **PROPPATCH** method, as specified in section 2.2.3.7; **MOVE** method, as specified in section 2.2.3.4; **COPY** method, as specified in section 2.2.3.1; **DELETE** method, as specified in [RFC2518] section 8.6; and **MKCOL** method, as specified in section 2.2.3.3.
- A WebDAV client can use the **resourcetag** XML element on a resource to detect whether it has already obtained the latest version of a specific resource.
- A WebDAV client can use the **resourcetag** XML element on a resource to ensure consistency when it uploads or downloads data.

2.2.2.7 resourcetaglist

Namespace: http://schemas.microsoft.com/repl/

Syntax: <!ELEMENT resourcetaglist (resourcetag+) >

The **resourcetaglist** XML element is a container for **resourcetag** XML elements, as specified in section 2.2.2.6.

2.2.3 Methods

2.2.3.1 COPY

The **COPY** method, as specified in [RFC2518], is used to duplicate an existing WebDAV resource. In the context of the WebDAV Extensions for Replication, the **COPY** method is used to copy a particular WebDAV resource.

A WebDAV replication–compliant server might not return the **resourcetag** XML element, as specified in section 2.2.2.6, as a result of the execution of a **COPY** operation, for example when copying a message to the mail submission URL, as specified in [MS-XWDMAIL].

2.2.3.2 GET

A client can use the **GET** method, as specified in [RFC2518], to fetch the contents of an existing WebDAV resource. In the context of the WebDAV Extensions for Replication, the **GET** method is used to download the content change for a particular WebDAV resource.

Every WebDAV replication–compliant server MUST return the updated **resourcetag** XML element, as specified in section 2.2.2.6, of the affected WebDAV resource in the response headers.

2.2.3.3 MKCOL

The **MKCOL** method, as specified in [RFC2518], is used to add a new collection resource to an existing collection resource. In the context of the WebDAV Extensions for Replication, the **MKCOL** method is used to upload the creation of a new collection resource.

Every WebDAV replication–compliant server MUST return the updated **resourcetag** element, as specified in section 2.2.2.6, and **repl-uid** element, as specified in section 2.2.2.5, of the affected collection resource in the response headers.

2.2.3.4 MOVE

The **MOVE** method, as specified in [RFC2518], is used to either move or rename an existing WebDAV resource. In the context of the WebDAV Extensions for Replication, the **MOVE** method is used to move or rename a particular WebDAV resource.

If the server changes the value of the **repl-uid** XML element, as specified in section 2.2.2.5, of the object, it MUST return a **Repl-uid:** header with the new value of the **repl-uid** XML element.

2.2.3.5 POST

The **POST** method, as specified in [RFC2518], is used to add a new non-collection resource to an existing collection by using a server-defined name. In the context of the WebDAV Extensions for Replication, the **POST** method is used to upload the contents of a new resource in a particular collection.

Every WebDAV replication–compliant server MUST return the **resourcetag** XML element, as specified in section 2.2.2.6, and **repl-uid** XML element, as specified in section 2.2.2.5, of the new non-collection resource in the response headers.

2.2.3.6 PROPFIND

The **PROPFIND** method, as specified in [RFC2518], is used to fetch the properties of an existing WebDAV resource. The **PROPFIND** method cannot be used to determine what items have changed for replication within a collection. However, this functionality is available through the **SEARCH** method, as specified in section 2.2.3.9.

2.2.3.7 PROPPATCH

The **PROPPATCH** method, as specified in [RFC2518], is used to set or remove the properties of an existing WebDAV resource. In the context of the WebDAV Extensions for Replication, the **PROPPATCH** method is used to upload the property changes for a particular WebDAV resource.

2.2.3.8 PUT

The **PUT** method, as specified in [RFC2518], is used to either add a new non-collection resource to an existing collection or update an existing non-collection resource. In the context of the WebDAV Extensions for Replication, the **PROPPATCH** method is used to upload the content change for a particular non-collection resource.

2.2.3.9 SEARCH

The **SEARCH** method, as specified in [MS-WDVSE], is used to search the properties of an existing WebDAV resource. In the context of the WebDAV Extensions for Replication, the **SEARCH** method is used to search for and download the property changes for WebDAV resources. The **SEARCH** method MUST be used to fetch the manifest of a collection or collection hierarchy, as specified in [RFC2518].

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The following sections specify extensions to the existing WebDAV commands, as specified in [RFC2518]. These commands SHOULD be processed as specified in [RFC2518], except in the cases specified in this section.

To keep a client view updated as changes happen on a server, a WebDAV replication client will typically perform a sequence of steps. First it will get an initial view of the underlying data by using a **SEARCH** command, as specified in [MS-WDVSE]. The server returns replication state information called a "collection **binary large object (BLOB)**" in the **collblob** XML element, as specified in section 2.2.2.2.

When the client wants to check for changes, it submits the same **SEARCH** command request, this time including the **collblob** XML element value that was returned by the server in the previous request. The server will return the results as a set of changes relative to the previous result set, omitting any unchanged resources.

If the client makes changes, it will receive a **resourcetag** XML element, as specified in section 2.2.2.6, that uniquely identifies the changes it made. It can include that **resourcetag** XML element in subsequent **SEARCH** command requests so that it does not have to retrieve its own changes an extra time.

3.1.5.1 COPY Method

A client can use the **COPY** method, as specified in [RFC2518], to move or rename a resource.

3.1.5.1.1 Version Checking

If the client has previously downloaded content or properties of a resource, the server MUST have returned the **resourcetag** XML element, as specified in section 2.2.2.6, of that particular resource. Under these circumstances, the client can include the **resourcetag** XML element in the request header of a **COPY** method, as specified in section 2.2.3.1, in the form of **If: (<resourcetag >)** or **If: (<repl-uid>)**, as specified in [RFC2518] section 9.4.

The **If: (<resourcetag>)** or **If: (<repl-uid>)** condition allows for client-initiated conflict detection.

3.1.5.1.2 Server-Side Modifications

The **COPY** method, as specified in [RFC2518], might trigger some server-side action that results in successful overwrite from the client perspective but modifications or transformations on the server side that result in a content and/or properties mismatch between the client and server. In this case, the server MUST return the new status code 210 Content Different. The response SHOULD also include information about what was affected during the execution of the **COPY** method on the server.

To solve this mismatch problem, the client might need to refetch the contents and/or properties of all the affected resources by using the **GET** and **PROPFIND** methods, as specified in [RFC2518].

3.1.5.1.3 Preventing Inadvertent Overwrite of an Existing Resource

The client can check to determine whether the resource it is intending to copy already exists at the destination. If the resource does exist, the client might not want to overwrite the existing resource. In this case, the client MUST include the **Overwrite: F** request header in the **COPY** request, as specified in [RFC2518], so as to avoid overwriting an existing resource.

3.1.5.1.4 Client-Initiated Conflict Detection

The client SHOULD include the **If: (<resourcetag>)** or **If: (<repl-uid>)** request header for the source collection, source non-collection, and destination collection in the **COPY** request, as specified in [RFC2518], and move the resource on the server only if the version matches. If the condition fails, the server MUST return the 412 Precondition Failed error code, as specified in [RFC2068] section 10.4.13.

3.1.5.2 GET Method

If the client issues a **GET** method request, as specified in [RFC2518], without any headers specific to replication, the response from the server will have the default behavior specified by [RFC2068] except that a WebDAV replication–compliant server MUST return the **resourcetag** XML element, as specified in section 2.2.2.6, of the affected resource.

3.1.5.2.1 Version Checking

If the client has previously downloaded content or properties of a resource, the server MUST have returned the **resourcetag** XML element, as specified in section 2.2.2.6, of that particular resource. The client can include the **resourcetag** XML element in the request header of a **GET** method, as specified in [RFC2518], in the form of **If: (<resourcetag>)** or **If: (<repl-uid>)**.

The **If: (<resourcetag>)** or **If: (<repl-uid>)** condition allows for client-initiated conflict detection.

3.1.5.2.2 Client-Initiated Conflict Detection

The client SHOULD include the **If: (<resourcetag>)** or **If: (<repl-uid>)** request header in the **GET** method request, as specified in [RFC2518], and fetch the resource on the server. If the condition fails, the server MUST return the 412 Precondition Failed error code, as specified in [RFC2068] section 10.4.13.

3.1.5.3 MKCOL Method

A client can use the **MKCOL** method, as specified in [RFC2518], to upload the creation of a new collection resource. A collection cannot be made at the Request-URI until one or more intermediate collections have been created. The server MUST NOT create those intermediate collections automatically.

If client issues a **MKCOL** method request without any headers that are specific to replication, the request will have the default behavior except that a WebDAV replication–compliant server MUST return the **resourcetag** XML element, as specified in section 2.2.2.6, of the affected resource. If the client tries to re-create a collection that already exists, the **MKCOL** method will fail with the 409 Conflict error code, as specified in [RFC2068] section 10.4.10.

3.1.5.4 MOVE Method

3.1.5.4.1 Version Checking

If the client has previously downloaded the content or properties of a resource, the server MUST have returned the **resourcetag** XML element, as specified in section 2.2.2.6, of that particular resource. Under these circumstances, the client can include the **resourcetag** XML element in the request header of a **MOVE** method, as specified in [RFC2518], in the form of **If: (<resourcetag>)** or **If: (<repl-uid>).**

The **If: (<resourcetag>)** or **If: (<repl-uid>)** condition allows for client-initiated conflict detection.

3.1.5.4.2 Server-Side Modifications

A **MOVE** method, as specified in [RFC2518], can trigger some server-side action that results in successful overwrite from the client perspective but modifications or transformations on the server side that result in a content and/or properties mismatch between the client and server.

In this case, the server MUST return the new status code, 210 Content Different. The response SHOULD also include information about what was affected during the execution of the **MOVE** method on the server.

In order to solve this mismatch problem, the client might need to refetch the contents and/or properties of all the affected resources by using the **GET** and **PROPFIND** methods, as specified in [RFC2518].

3.1.5.4.3 Preventing Inadvertent Overwrite of an Existing Resource

The client might want to check to determine whether the resource it is intending to move already exists at the destination. If so, it might not want to overwrite the existing resource. In this case, the client MUST include the **Overwrite: F** request header in the **MOVE** method request, as specified in [RFC2518], to avoid overwriting an existing resource.

3.1.5.5 POST Method

If a client issues a **POST** method request, as specified in [RFC2518], without any headers that are specific to replication, the request will have the default behavior specified in [RFC2068] and [RFC2518], except that a WebDAV replication–compliant server MUST return the **resourcetag** XML element, as specified in section 2.2.2.6, of any created or updated resource.

3.1.5.6 PROPFIND Method

The client can fetch the **http://schemas.microsoft.com/repl/resourcetag** property of every WebDAV resource reported in the response of the **PROPFIND** method, as specified in [RFC2518].

3.1.5.7 PROPPATCH Method

The behavior of the **PROPPATCH** method, as specified in [RFC2518], is very similar to the behavior of the **PUT** method, as specified in [RFC2518], except that the **PROPPATCH** method deals with properties rather than the resource contents.

3.1.5.8 PUT Method

If the client issues a **PUT** request, as specified in [RFC2518], without any headers that are specific to replication, the request will have the default behavior as specified by [RFC2068] and [RFC2518].

3.1.5.8.1 Version Checking

If the client has previously downloaded the content or properties of a resource, the server MUST have returned the **resourcetag** XML element, as specified in section 2.2.2.6, of that particular resource. The client can include the **resourcetag** XML element in the request header of a **PUT** method as specified in [RFC2518], in the form of **If: (<resourcetag>)**. The client can include the **repl-uid** XML element, as specified in section 2.2.2.5, in the request header of a **PUT** method in the form of **If: (<repl-uid>)**.

The **If: (<resourcetag>)** or **If: (<repl-uid>)** condition allows for client-initiated conflict detection.

3.1.5.8.2 Server-Side Modifications

The **PUT** method, as specified in [RFC2518], can trigger some server-side action that results in successful overwrite from client perspective but modifications or transformations on the server side that result in a content and/or properties mismatch between the client and server. Because every **PUT** method MUST return the updated **resourcetag** XML element, as specified in section 2.2.2.6, there is a mismatch between the content and/or properties on the client and the content and/or properties that are reflected by the **resourcetag** XML element.

In this case, the server MUST return the new status code, 210 Content Different. The response SHOULD also include information about what was affected during the execution of the **PUT** method on the server.

To solve this mismatch problem, the client might need to refetch the contents and/or properties of the affected resource by using the **GET** and **PROPFIND** methods, as specified in [RFC2518].

3.1.5.8.3 Preventing Inadvertent Overwrite of an Existing Resource

Sometimes the client might want to check to determine whether the resource it is intending to use the **PUT** method on already exists, and if so, it might not want to overwrite the existing contents.

In this case, the client SHOULD include the **If-None-Match** header with value "*", as specified in [RFC2068] section 14.26, in the **PUT** method request to avoid overwriting an existing resource.

3.1.5.8.4 Client-Initiated Conflict Detection

The client SHOULD include the **If: (<resourcetag>)** or **If: (<repl-uid>)** request header in the **PUT** method request, as specified in [RFC2518], and update the resource on the server only if the version matches. If the condition fails, the server MUST return a 412 Precondition Failed error code, as specified in [RFC2518] section 8.8.5.

3.1.5.9 SEARCH Method

The client can include the **collblob** XML element, as specified in section 2.2.2.2, to retrieve only the changes relative to a previous **SEARCH** method request, as specified in [MS-WDVSE].

The client can explicitly fetch the http://schemas.microsoft.com/repl/resourcetag property of every WebDAV resource reported in the response of the **SEARCH** method.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 collblob Element

The **collblob** XML element, as specified in section 2.2.2.2, is an encoded string that specifies the state of a collection, as specified in [RFC2518]. It is generated by the server, and the client MUST treat it as an opaque value. It can be sent to the server to communicate the state of a collection that the client has previously retrieved, and a new value is returned by the server after various operations.

3.2.5.2 changetype Element

The WebDAV Extensions for Replication define the following attributes of the **changetype** XML element, as specified in section 2.2.2.1:

- **change** – An existing resource has been updated. The client MUST assume that the default attribute of the **changetype** XML element in the manifest response from the server is **change,** if no **changetype** XML element is given in the **response** XML element.
- **delete** – A resource has been deleted. A server MUST return the **repl-uid** property, as specified in section 2.2.2.5, of the resource if the attribute of the **changetype** XML element on the resource is **delete**.
- **new** – A new resource has been added since the last time the **collblob** XML element, as specified in section 2.2.2.2, was returned.
- **read** – The only thing that has changed on the resource is the read/unread state. If the read/unread state has changed and any other change has occurred, the attribute of the **changetype** XML element will be **change**.

The protocol does not prevent the extensibility in terms of the other potential **changetype** element attributes based on the client-server negotiation.

The change in the manifest applies only to the resource referenced in the **DAV:href** property, as specified in [MS-XWDMAIL] section 2.2.2.1.4, in the response XML element of the manifest.

In the absence of any other additional **changetype** XML elements, the following occurs:

- A move operation SHOULD be indicated by the server in the source collection with the **delete** attribute on the **changetype** XML element and in the destination collection manifest with the **change** attribute in the **changetype** XML element.
- A copy operation SHOULD be indicated by the server with a **change** attribute on the **changetype** XML element in the destination collection manifest.
- A rename operation and a move operation SHOULD be treated as the same type of operation for the purposes of registering the type of the change on a resource.

3.2.5.3 contenttag Element

Every time the contents or properties of resources within the collection change, the **http://schemas.microsoft.com/repl/contenttag** property on the collection MUST be updated to reflect the change. A server that implements WebDAV replication MUST support the **http://schemas.microsoft.com/repl/contenttag** property on every WebDAV collection resource that can be replicated. The server MUST guarantee that the values of two **contenttag** XML elements are binary comparable. The client can store the values of the **contenttag** XML elements for future comparisons. The **contenttag** XML element can do the following:

- Provide an efficient way for the client to determine whether a collection has changed since the last time it synchronized by comparing the value on the client. The client SHOULD use the **PROPFIND** or **SEARCH** methods, as specified in [RFC2518], to fetch the **http://schemas.microsoft.com/repl/contenttag** property of the collection and then compare the value against its previously obtained value from the server.
- Allow an efficient and easy way to check for not only collection hierarchy changes but also collections for which the contents have changed. The client SHOULD use the **PROPFIND** method or the **SEARCH** method to fetch the **http://schemas.microsoft.com/repl/contenttag**

property of the collections in the hierarchy and then compare it against its previously obtained values from the server without the **collblob** XML element, as specified in section 2.2.2.2, or the **repl** XML element, as specified in section 2.2.2.5.

- In conjunction with the **repl-uid** XML element, MAY allow the client find out whether a collection has moved.
- Be used in an **If** header, as specified in [RFC2518] section 9.4, to make sure that an operation (especially **DELETE**, as specified in [RFC2518] section 8.6) on a collection will only happen if the contents of the collection have not changed. As such, it can be used in an **If** header anywhere that the **resourcetag** XML element, as specified in section 2.2.2.6, or the **repl-uid** XML element, as specified in section 2.2.2.5, is allowed.

3.2.5.4 repl-uid Element

A server that implements WebDAV replication MUST support the http://schemas.microsoft.com/repl/repl-uid property on every replicated WebDAV resource.

The http://schemas.microsoft.com/repl/repl-uid property MAY be obtained as property on a resource by using the **PROPFIND** method command or the **SEARCH** method command, as specified in [RFC2518].

Delete operations might not have valid URL identifiers, but they MUST have valid values in the **repl-uid** XML elements, as specified in section 2.2.2.5.

A client can include the unique identifier in the request header only if its intention is to ensure that it is dealing with the same resource that it has always known.

Note that a server can change the http://schemas.microsoft.com/repl/repl-uid property on a resource, if the resource is moved, renamed, or copied.

The server MUST return the **repl-uid** XML element of the resource as a response header in every **PUT**, **POST**, **MKCOL**, and **PROPPATCH** method request, as specified in [RFC2518].

A server MUST ignore any request headers related to the **repl-uid** element for a **POST** method request because they do not hold any special meaning or purpose.

3.2.5.5 resourcetag Element

A server that implements WebDAV replication MUST be able to generate the **resourcetag** XML element, as specified in section 2.2.2.6. The server also MUST provide support for the http://schemas.microsoft.com/repl/resourcetag property on every replicated WebDAV resource. The value of this property is a URI. Note that the contents of the **resourcetag** XML element are opaque to the client. A server MUST ignore any request headers related to the **resourcetag** XML element for a **POST** method request because they do not hold any special meaning or purpose. A WebDAV replication–compliant server MUST NOT return the **resourcetag** element of the affected resources in the response headers for a **PROPPATCH** method and a **PUT** method due to the possibility of a large result set.

A **resourcetag** XML element MUST meet the following requirements:

- Two **resourcetag** XML elements MUST be binary comparable by the client.
- The server MUST guarantee that if two **resourcetag** XML elements are the same when compared, the resource MUST be the same.

- The client MUST be able to fetch the http://schemas.microsoft.com/repl/resourcetag property on the resource.
- It MUST be possible for the client to include the **resourcetag** XML element or **repl-uid** XML element, as specified in section 2.2.2.5, in the **If:** request header of any WebDAV request.
- The server MUST return the **resourcetag** XML element of the resource as a response header in every **GET**, **PUT**, **POST**, **MKCOL**, **PROPPATCH**, and **DELETE** method request, as specified in [RFC2518].

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Client Has Never Fetched the Manifest of a Collection

A client has never fetched the manifest for a collection. The client includes the **searchrequest** element, as described in [MS-WDVSE]; **repl** element, as described in section 2.2.2.4; and **collblob** element, as described in section 2.2.2.2, to request the manifest.

```
>>Request
SEARCH /private/user0/inbox HTTP/1.1
Host: www.company.com
Content-type: text/xml
Content-length: {insert length here}

<?xml version="1.0"?>
<D:searchrequest xmlns:D="DAV:"
    xmlns:R="http://schemas.microsoft.com/repl/"
    xmlns:M="urn:schemas:mail: ">
    <R:repl>
     <R:collblob/>
    </R:repl>
    <D:sql>
     SELECT 'urn:schemas:mail:Size', 'urn:schemas:mail:Importance',
               'http://schemas.microsoft.com/repl/resourcetag'
          FROM SCOPE ('SHALLOW TRAVERSAL OF "http://www.company.com/private/user0/inbox"')
    </D:sql>
</D:searchrequest>

>>Response
HTTP/1.1 207 Multi-Status
Content-type: text/xml
Content-length: {insert length here}

<?xml version="1.0"?>
<D:multistatus xmlns:D="DAV:"
    xmlns:R="http://schemas.microsoft.com/repl/"
    xmlns:M="urn:schemas:mail:">
     <R:repl>
          <R:collblob>clientopaquedata</R:collblob>
     </R:repl>
     <D:response>
                  <D:href>http://www.company.com/private/user0/inbox/msg1</D:href>
                  <D:propstat>
                      <D:status>HTTP/1.1 200 OK</D:status>
                      <D:prop>
                          <D:Size>1000</D:Size>
                          <M:Importance>High</M:Importance>
                    <R:resourcetag>doc1-01</R:resourcetag>
                      </D:prop>
                  </D:propstat>
                  <R:changetype>change</R:changetype>
          </D:response>
          <D:response>
                  <D:href>http://www.company.com/private/user0/inbox/msg4</D:href>
                  <D:propstat>
                      <D:status>HTTP/1.1 200 OK</D:status>
                      <D:prop>
                          <D:Size>14400</D:Size>
```

```
                        <M:Importance>High</M:Importance>
                    <R:resourcetag>doc2-02</R:resourcetag>
                      </D:prop>
                  </D:propstat>
                  <R:changetype>change</R:changetype>
      </D:response>
  </D:multistatus>
```

4.2 Client-Side Detection to Avoid Unnecessary Downloads

A client has fetched the manifest and the **collblob** element, as specified in section 2.2.2.2, for a collection **doccoll** before and is seeking an updated value for the **collblob** element and manifest for the collection.

```
>>Request
SEARCH /doccoll HTTP/1.1
Host: www.company.com
Content-type: text/xml
Content-length: {insert length here}

<?xml version="1.0"?>
<D:searchrequest xmlns:D="DAV:"
    xmlns:R="http://schemas.microsoft.com/repl/"
    xmlns:M="urn:schemas:mail:">
    <R:repl>
<R:collblob>clientopaquedata</R:collblob>
    </R:repl>
</D:searchrequest>

>>Response
HTTP/1.1 207 Multi-Status
Content-type: text/xml
Content-length: {insert length here}

<?xml version="1.0"?>
<D:multistatus xmlns:D="DAV:"
    xmlns:R="http://schemas.microsoft.com/repl/"
    xmlns:M="urn:schemas:mail:">
<R:repl>
<R:collblob>clientopaquedata</R:collblob>
</R:repl>
<D:response>
        <D:href>http://www.company.com/doccoll/msg1</D:href>
        <D:propstat>
            <D:status>HTTP/1.1 200 OK</D:status>
            <D:prop>
                <D:Size>1000</D:Size>
                <M:Importance>High</M:Importance>
<R:resourcetag>rt:doc1-01</R:resourcetag>
            </D:prop>
        </D:propstat>
        <R:ChangeType>change</R:ChangeType>
    </D:response>
    <D:response>
        <D:href>http://www.company.com/doccoll/msg4</D:href>
        <D:propstat>
```

```
            <D:status>HTTP/1.1 200 OK</D:status>
            <D:prop>
                <D:Size>14400</D:Size>
                <M:Importance>High</M:Importance>
<R:resourcetag>rt:doc1-01</R:resourcetag>
            </D:prop>
        </D:propstat>
        <R:ChangeType>change</R:ChangeType>
</D:response>
</D:multistatus>
```

While the client was offline, a document named docE in the collection **doccoll** was updated.

The client uses a Web browser to download the document named docE. The server returns the contents of the document docE and its corresponding resource URI in the **Resourcetag** header, as follows:

```
>>Request
GET /doccoll/docE HTTP/1.1

>>Response
HTTP/1.1 200 OK
Resourcetag: <rt:19a23000c26511d18faf00600892444c>
Content-type: text/plain
Content-length: {insert length here}

This is the content of text document docE.
```

The client saves the contents of the document docE, and its resource URI.

After a while, the user wants to find out what has changed since the last time it synchronized with the same server. The client sends a request for the manifest and **resourcetag** property for the **doccoll** collection by including its previous replication state in the **collblob** XML element in the request, and the server responds with a manifest that includes the change that corresponds to docE.

To avoid unnecessarily re-downloading documents, the client compares the **resourcetag** property that it obtained as part of the manifest with the **resourcetag** property that it persisted before for document docE and finds that it already has the latest version of the document docE.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® Exchange Server 2003
- Microsoft® Exchange Server 2007

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

7 Change Tracking

This section identifies changes that were made to the [MS-XWDREPL] protocol document between the November 2010 and March 2011 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1 Introduction	Removed normative reference citations.	N	Content removed for template compliance.
3.1.5.5 POST Method	Moved server behavior into Server Details section.	N	Content updated.
3.1.5.6 PROPFIND Method	Moved server behavior into Server Details section for the resourcetag element.	N	Content updated.
3.1.5.7 PROPPATCH Method	Removed server behavior.	N	Content updated.
3.1.5.8 PUT Method	Removed server behavior.	N	Content updated.
3.1.5.9 SEARCH Method	Removed server behavior.	N	Content updated.
3.2.5.4 repl-uid Element	Added server behavior from Client Details POST Method section.	N	Content updated.
3.2.5.5 resourcetag Element	Added server behavior from Client Details POST Method and PROPFIND method sections.	N	Content updated.

8 Index

N

O

P

R

S

T

V

X