[MS-WDV]: Web Distributed Authoring and Versioning (WebDAV) Protocol: Client Extensions

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
05/11/2007	0.1		MCPP Milestone 4 Initial Availability
08/10/2007	0.1.1	Editorial	Revised and edited the technical content.
09/28/2007	0.1.2	Editorial	Revised and edited the technical content.
10/23/2007	0.1.3	Editorial	Revised and edited the technical content.
11/30/2007	0.1.4	Editorial	Revised and edited the technical content.
01/25/2008	0.1.5	Editorial	Revised and edited the technical content.
03/14/2008	1.0	Major	Updated and revised the technical content.
05/16/2008	1.0.1	Editorial	Revised and edited the technical content.
06/20/2008	2.0	Major	Updated and revised the technical content.
07/25/2008	2.0.1	Editorial	Revised and edited the technical content.
08/29/2008	3.0	Major	Updated and revised the technical content.
10/24/2008	4.0	Major	Updated and revised the technical content.
12/05/2008	4.0.1	Editorial	Revised and edited the technical content.
01/16/2009	4.0.2	Editorial	Revised and edited the technical content.
02/27/2009	5.0	Major	Updated and revised the technical content.
04/10/2009	5.0.1	Editorial	Revised and edited the technical content.
05/22/2009	6.0	Major	Updated and revised the technical content.
07/02/2009	6.0.1	Editorial	Revised and edited the technical content.
08/14/2009	6.0.2	Editorial	Revised and edited the technical content.
09/25/2009	6.1	Minor	Updated the technical content.
11/06/2009	6.1.1	Editorial	Revised and edited the technical content.
12/18/2009	7.0	Major	Updated and revised the technical content.
01/29/2010	7.0.1	Editorial	Revised and edited the technical content.
03/12/2010	7.0.2	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
04/23/2010	7.0.3	Editorial	Revised and edited the technical content.
06/04/2010	7.0.4	Editorial	Revised and edited the technical content.
07/16/2010	8.0	Major	Significantly changed the technical content.
08/27/2010	9.0	Major	Significantly changed the technical content.
10/08/2010	9.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	9.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	9.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	9.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction

The HTTP Extensions for Distributed Authoring—WEBDAV Protocol (WebDAV), as specified in [RFC4918], extends the standard **Hypertext Transfer Protocol (HTTP)** mechanisms that are specified in [RFC2616] to provide file access and content management over the Internet. The WebDAV Protocol enables an Internet-based file system. However, some types of files—for example, files with programmatically-derived content—are not easily managed by WebDAV Protocol. Also, some protocol interactions—for example, the separation of **properties** and content—are less than optimal for file system usage.

The client extensions in this specification, Web Distributed Authoring and Versioning (WebDAV) Protocol: Client Extensions, extend the WebDAV Protocol, as specified in [RFC4918], by introducing new headers that both enable the file types that are not currently manageable and optimize protocol interactions for file system clients. These WebDAV Protocol: Client Extensions do not introduce new functionality into the WebDAV Protocol, but instead optimize processing and eliminate the need for special-case processing.

1.1 Glossary

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

Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
Secure Sockets Layer (SSL)
Transport Layer Security (TLS)

The following terms are specific to this document:

entity: Any document on a server that is accessible by using a **Hypertext Transfer Protocol (HTTP)** URL.

file browsing: A process of viewing or searching document collections.

locking: A mechanism that is used for overwrite protection. **Locking** may be applied to individual **resources** or to entire collection hierarchies. This term is used as specified in [RFC4918] sections 6 and 7.

property: A name/value pair that associates metadata with a **resource**. This term is used as specified in [RFC4918] section 4.

resource: An **entity** that can be identified by a URI. This term is used as specified in [RFC2616] section 1.3.

thicket: A group of supporting files and folders on the **Web server** that together store the content of one logical document.

WebDAV: Web Distributed Authoring and Versioning Protocol (WebDAVs, as specified in [RFC4918].

WebDAV client: A computer that uses the WebDAV Protocol, as specified in [RFC4918], to retrieve data from the **WebDAV server**.

WebDAV server: A computer that supports the WebDAV Protocol, as specified in [RFC4918]. **WebDAV clients** can connect to and retrieve data from a **WebDAV server**.

Web server: A computer on the Web that is connected to the Internet backbone and stores Web pages that can be retrieved by a client.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as specified 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.

[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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

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

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, http://www.ietf.org/rfc/rfc3986.txt

[RFC4918] Dusseault, L, Ed., "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", RFC 4918, June 2007, http://www.ietf.org/rfc/rfc4918.txt

1.2.2 Informative References

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

[MSASP] Microsoft Corporation, "Active Server Pages", http://msdn.microsoft.com/en-us/library/aa286483.aspx

[MSDN-STC] Microsoft Corporation, "Storage Technologies Collection", March 2003, http://technet2.microsoft.com/WindowsServer/en/Library/616e5e77-958b-42f0-a87f-ba229ccd81721033.mspx

1.3 Overview

The WebDAV Protocol is a set of methods, headers, and content-types that extend the Hypertext Transfer Protocol -- HTTP/1.1, as specified in [RFC2616]. The WebDAV Protocol allows data to be written to Internet servers and is an Internet standard for collaborative authoring, as specified in [RFC4918].

The WebDAV Protocol expands the basic support in HTTP/1.1 for content authoring by introducing additional methods and headers that provide support for **resource** properties and other base functions, such as resource **locking**. These new capabilities make the WebDAV Protocol suitable for basic remotely mountable file systems.

WebDAV Protocol: Client Extensions specifies the following extensions to the base WebDAV Protocol:

- A mechanism, which is based on the WebDAV Protocol and HTTP/1.1, to indicate support for the extensions that are covered in this document.
- A header to indicate if an **Entity** is to be returned as is or if any associated programmatic processing should be performed and the result returned.
- An extension that provides a way to GET and PUT properties along with Entity content, offering more efficient **file browsing**.
- A header that enables the bundling of locking information by using GET, PUT, and POST commands to improve the efficiency of locking semantics.

WebDAV Protocol: Client Extensions also specifies the following extension to the base HTTP protocol:

- A mechanism for an HTTP request to be retried using the HTTP 449 status code extension.

1.4 Relationship to Other Protocols

WebDAV Protocol: Client Extensions rely on the HTTP Extensions for Distributed Authoring—WEBDAV, as specified in [RFC4918], which in turn relies on HTTP/1.1, as specified in [RFC2616]. WebDAV Protocol: Client Extensions also rely on the **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as specified in HTTP Over TLS, [RFC2818], for data protection services.

1.5 Prerequisites/Preconditions

WebDAV Protocol: Client Extensions require a **WebDAV server**, as specified in [RFC4918], that supports the OPTIONS command.

This specification also requires that **WebDAV clients** have URLs that point to WebDAV servers.

1.6 Applicability Statement

This protocol is applicable in scenarios that require efficient file operations. Note that this document specifies only those extensions that are needed to enable efficient file system clients. These extensions do not add any additional functionality. Instead, they help reduce the network traffic and increase the performance of clients that use the WebDAV Protocol, as specified in [RFC4918].

1.7 Versioning and Capability Negotiation

This document introduces no new versioning mechanisms except those that already exist in the WebDAV Protocol and HTTP/1.1.

Negotiation of the WebDAV Protocol and of HTTP/1.1 options in general is specified in [RFC4918] and [RFC2616], respectively. The X-MSDAVEXT header is used as part of the HTTP/1.1 OPTIONS discovery mechanism to indicate WebDAV server support for this specification.

1.8 Vendor-Extensible Fields

The extensions that are defined in this protocol can be extended in constrained ways, as specified in section 2.2.3.

1.9 Standards Assignments

No standards body has approved or governs this document or its header names and values. This specification conforms to the form and behavior of other custom HTTP headers, as specified in [RFC2616] section 4.2.

2 Messages

2.1 Transport

Messages are transported by using HTTP, as specified in [RFC4918] and [RFC2616].

This protocol MAY be used with **Secure Sockets Layer (SSL)** or **Transport Layer Security (TLS)**, as specified in [RFC2246].<1>

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>

2.2 Message Syntax

The extension headers in this protocol conform to the form and behavior of other custom HTTP headers, as specified in [RFC2616] section 4.2. They are consistent with the WebDAV Protocol verbs and headers, as specified in [RFC4918] sections 8 and 9.

The following header extensions are specified in this section:

- The X-MSDAVEXT response header has been added to the OPTIONS response and indicates support for the extensions that are covered in this document, as specified in section 2.2.1.
- The Translate request header allows the client to request the source of an entity, as specified in section 2.2.2.
- The X-MSDAVEXT_ERROR response header provides a mechanism for extended error handling, as specified in section 2.2.3.
- The X-MSDAVEXTLockTimeout request/response header enables bundling of LOCK information (as specified in [RFC4918] section 8.10) with GET, PUT, and POST messages, as specified in section 2.2.4.1.
- The Lock-Token request/response header enables bundling of LOCK information with GET, PUT, and POST messages, as specified in section 2.2.4.2.
- The Ms-Echo-Request header is sent by the server when returning a 449 Retry With status, as specified in section 2.2.7.
- The Ms-Echo-Reply header is sent by the client when making a request in response to a 449 Retry With status, as specified in section 2.2.8.

The following new content type is specified in this extension:

- The Multipart/MSDAVEXTPrefixEncoded content type allows entity properties and entity content to be bundled in a single message, as specified in section 2.2.5.

The following new status code is specified in this extension:

- The 449 Retry With status code allows the server to indicate that the request did not contain sufficient information and should be retried by the client, as specified in section 2.2.6.

2.2.1 WebDAV Extension Header

The X-MSDAVEXT header has been added to indicate support for the WebDAV Protocol: Client Extensions and to request optional server behavior.

This new header is defined as follows (using the Augmented Backus-Naur Form (ABNF) Syntax as specified in [RFC2616] section 2.1):<3>

```
MS-WebDAV Extension Header = "X-MSDAVEXT" ":" Ext-options
Ext-options = "1" | "PROPFIND" | "PROPPATCH"
```

The presence of this header with a value of 1 in the OPTIONS response indicates that the server supports the new extensions described in this document. This header with a value of 1 is valid only on an OPTIONS response.

The presence of this header with a value of PROPFIND on a GET or POST request indicates that the server which sends X-MSDAVEXT: 1 in its OPTIONS response MUST return both the properties and the actual file together in the response using the multipart/MSDAVEXTPrefixEncoded content type, as specified in section 2.2.5. In accordance with [RFC2616] section 9.4, a server MUST respond to a HEAD request with this header with the same headers it would in a GET request but without the message body.

The presence of this header with a value of PROPPATCH on a PUT request indicates that the body of the request contains both the properties and the actual file together in the request. The content type header MUST indicate the media type of multipart/MSDAVEXTPrefixEncoded.

This header SHOULD be ignored on all other commands and for all other values.<4>

2.2.2 Translate Header

Many resources that are obtained from a **Web server** are returned exactly as is. However, some resources are programmatically interpreted by the Web server, and the result of that interpretation is returned instead of the source representation. For example, without a mechanism to control programmatic interpretation, a request to retrieve an Active Server Page (ASP) from the Web server returns the processed HTML file rather than the actual source of the ASP page. For more information, see [MSASP].

The WebDAV Protocol: Client Extensions introduce a new Translate header so that a WebDAV client can indicate what representation it wants. This header indicates to the Web server whether or not it should perform translation, that is, programmatic interpretation, of the file.

This new header is defined as follows by using the augmented Backus-Naur Form (BNF) syntax, as specified in [RFC2616] section 2.1.

```
Translate-header = "Translate" ":" Translate-value
Translate-value = "t" | "f" | "F"
```

If Translate-value is "t", the server is to process the content before returning it to the client; if it is "f" | "F", the server is to return the unprocessed content (that is, the source without the programmatic interpretation) to the client. The values "f" and "F" are synonymous. All other values SHOULD be ignored by the server. For more information about the processing of this header, see section 3.2.5.1.<5>

This header MUST be supported on the GET verb and MAY be supported on other verbs.<6>

2.2.3 Extended Error Handling

The current errors that are returned by the HTTP protocol are not sufficient to support all the error conditions that occur in file handling. This section specifies a mechanism to extend HTTP error handling by using the X-MSDAVEXT_ERROR header.

Extended error handling MAY be used by applications in order to provide more specific information to the application user for an error that occurred on the server. The X-MSDAVEXT_ERROR header, which is returned by the WebDAV server, MAY be included in any WebDAV server response.<7>

This new header uses the Augmented Backus-Naur Form (ABNF) syntax, as specified in [RFC2616] section 2.1, and is defined as follows:

```
MSError-Header = "X-MSDAVEXT_ERROR" ":" Extended-error "; " Error-string
Extended-error = 1*DIGIT
Error-string = 1*TEXT
```

An Extended-error is an implementation-specific number that provides additional information about the cause of the HTTP error.

An Error-string is a percentage-encoded UTF-8 string, as specified in [RFC3986] section 2.1, that gives additional explanatory text about the cause of the error. This string is not significant to protocol operation and is intended only for display and logging purposes.

2.2.4 Lock Headers

WebDAV Protocol: Client Extensions extend the semantics of an existing LOCK header to enable resource locking and unlocking capabilities on the GET, PUT, and POST commands. This enhancement eliminates the need to send separate messages. This protocol also adds the X-MSDAVEXTLockTimeout header to indicate the lock duration. The Lock-Token header indicates the lock token, as specified in [RFC4918] section 6.3.<8>

2.2.4.1 Lock Time-out Header

The Lock Time-out header, which uses the augmented Backus-Naur Form (BNF), as specified in [RFC2616] section 2.1, is defined as follows:

```
MS-LockTimeout Header = "X-MSDAVEXTLockTimeout" ":" 1#TimeInterval
TimeInterval = ("Second-" TimeOutVal | "Infinite")
TimeOutVal = 1*DIGIT
```

The value of this header is interpreted identically to the Timeout header described in [RFC4918] section 10.7.

The Lock Time-out header is valid on the requests and responses for GET, PUT, and POST with the following meanings:

- On a request: The number portion of this header specifies the time-out for creating or refreshing the lock. A value of 0 indicates an unlock request.
- On a response: When included in a response to create or refresh the lock operation, the Lock Time-out header specifies the remaining time for which the lock is valid.

The Lock-Token header MUST be included to unlock or refresh the file, as specified in sections 3.1.5.4 and 3.2.5.2.

2.2.4.2 Lock-Token Header

The Lock-Token header in the WebDAV Protocol: Client Extensions is the same Lock-Token header as specified in [RFC4918] section 10.5. This extension extends only the applicability of the header.

The Lock-Token header MAY be included on GET, PUT, and POST requests and responses. When this header and the X-MSDAVEXTLockTimeout header are included on these requests, it instructs the server to either refresh or release the lock; or to perform a write operation on a locked object, as specified in section 3.1.5.4.<9>

The Lock-Token header SHOULD be included in a PUT request that acts on a locked resource. It MAY be validated by the server before honoring a PUT request. A server MUST NOT fail a PUT request simply because the header is not present in the request. The Lock-Token MUST be included in GET, HEAD, POST, and PUT responses from servers that return an X-MSDAVEXT: 1 header from an OPTIONS request.

The Lock-Token header MUST NOT be included on a GET or POST command without a lock time-out header.

2.2.5 Multipart Content Type

To enable efficient transfer of multiple sets of information in an HTTP request or response, this extension defines a new media type, multipart/MSDAVEXTPrefixEncoded, as specified in [RFC2616] section 3.7. When using this media type, the Content-Length header (as specified in [RFC2616] section 14.13) MUST include the length of the body that includes the size fields that are described below.

The entity body MUST be encoded as specified below.

```
<Properties-Size> <Properties> <File-Size> <File-contents>
```

Properties-Size: The size of the properties by using hexadecimal string representation. This representation of a 64-bit number MUST be a 16-byte string of hexadecimal digits that are prefixed with padding zeros where necessary.

Properties: The properties of the Resource. For a response, this MUST be the same as the body of the response, as specified in [RFC4918] section 9.1, for a PROPFIND command requesting all properties of the resource. For a request, this MUST be the same as the body of a PROPPATCH command, as specified in [RFC4918] section 9.2.

File-Size: The size of the file by using hexadecimal string representation. This representation of a 64-bit number MUST be a 16-byte string of hexadecimal digits that are prefixed with padding zeros where necessary.

File-Contents: The contents of the file. For a response, this MUST be the same as the body of a GET response, as specified in [RFC2616] section 9.3. For a request, this MUST be the same as the body of a PUT request, as specified in [RFC2616] section 9.6.<10>

2.2.6 449 Retry With Status Code

The 449 Retry With status code indicates that the request cannot be satisfied because insufficient information was provided by the client.

The new extension status code is defined as follows (using the Augmented Backus-Naur Form (ABNF) syntax, as specified in [RFC2616] section 2.1).

```
Status-code = "449"
Reason-phrase = "Reply With"
```

2.2.7 Ms-Echo-Request Header

The Ms-Echo-Request header is a response header. It is returned by the server when replying to a request with the 449 Retry With status code.

The new header is defined as follows (using the Augmented Backus-Naur Form (ABNF) syntax, as specified in [RFC2616] section 2.1).

```
Ms-Echo-Request Header = "Ms-Echo-Request" ":" field-value
```

The field-value rule is specified in [RFC2616] section 4.2. The field value of the Ms-Echo-Request header is opaque. Its value MUST NOT be examined by the client except as necessary to ensure its compliance with the client's implementation of the underlying HTTP protocol.

2.2.8 Ms-Echo-Reply Header

The Ms-Echo-Reply header is a request header that is sent by the client when reissuing a request for a resource after receiving a 449 Retry With status code for a previous request.

The new header is defined as follows (using the Augmented Backus-Naur Form (ABNF) syntax, as specified in [RFC2616] section 2.1).

```
Ms-Echo-Reply Header = "Ms-Echo-Reply" ":" field-value
```

The field-value rule is specified in [RFC2616] section 4.2. The field value of the Ms-Echo-Reply header is opaque. Its value MUST NOT be examined by the server except as necessary to ensure its compliance with the server's implementation of the underlying HTTP protocol.

3 Protocol Details

As specified in [RFC4918], the WebDAV Protocol operates between an initiator (a WebDAV client) and a responder (a WebDAV server). In this section, the client and the server behaviors for the WebDAV Protocol: Client Extensions are specified. This section also includes details on abstract data models, syntax, and message processing rules.

3.1 WebDAV Client Details

3.1.1 Abstract Data Model

A WebDAV client SHOULD maintain a logical Boolean variable for each server with which it communicates. This variable should be set to TRUE if the server has returned an X-MSDAVEXT: 1 header on an OPTIONS response, and FALSE otherwise.<11>

3.1.2 Timers

No new timers are required except those in the WebDAV Protocol, as specified in [RFC4918].

3.1.3 Initialization

Prior to general interaction with a WebDAV server, the WebDAV client SHOULD perform an OPTIONS request to determine the correct setting of the Boolean value for MSDAVEXT (if the extensions are supported).

3.1.4 Higher-Layer Triggered Events

No new events are triggered except those in the WebDAV Protocol, as specified in [RFC4918].

3.1.5 Processing Events and Sequencing Rules

WebDAV Protocol: Client Extensions can be used to combine multiple commands in a single request; to propagate extended error information between the client and the server; and to allow programmatic interpretation to be controlled by the client.

A WebDAV client SHOULD send an OPTIONS request to the server and query the headers in the server response for the presence of the X-MSDAVEXT: 1 header. This information SHOULD be used by the client when it sends requests to this server. The client MUST NOT add the new WebDAV Protocol extensions (as specified in this protocol) to the commands when it communicates with a server that responds to the OPTIONS command without specifying the X-MSDAVEXT: 1 header.

If the client does not support the extensions, the client MAY ignore the header and SHOULD NOT use the new **WebDAV** extensions (as specified in this protocol). If the server adds the optional headers, the headers SHOULD be ignored by the client as unrecognized.<12>

3.1.5.1 WebDAV Extension Header

The X-MSDAVEXT header has been added to indicate support for WebDAV Protocol: Client Extensions and to request optional server behavior. For more information, see section 3.1.5.

3.1.5.1.1 Extensions to GET and POST

File system clients need to obtain file properties together with the contents of the file.

An X-MSDAVEXT: PROPFIND header in the request from the client instructs the server to return the properties of the file together with the content of the file. If the server returns a content type that is not multipart/MSDAVEXTPrefixEncoded in the response, the client SHOULD treat this as an error.<13>

This header SHOULD be included only on GET commands that are targeted to servers that support these extensions, as advertised by an X-MSDAVEXT: 1 header in the OPTIONS command response from the server.<14>

3.1.5.1.2 Extensions to PUT

File system clients might need to update file properties along with the contents of the file.

An X-MSDAVEXT: PROPPATCH header in the request instructs the server that the properties of the file are included in the body together with the content of the file, and are to be updated together.

This header SHOULD be included only on PUT commands that are targeted to servers that support the extensions, as advertised by an X-MSDAVEXT: 1 header in the OPTIONS command response from the server.<15>

The Content-Type: multipart/MSDAVEXTPrefixEncoded header MUST be added by the client on the request, and the body SHOULD be of the format specified in section 2.2.5.

3.1.5.2 Translate Header

The WebDAV client MAY add the Translate header with a flag of "f" to a request if it needs the source of a file without any translation; otherwise, this header SHOULD be omitted. This header MAY be added to the commands that are sent to a WebDAV server that does not advertise the extensions.<16>

3.1.5.3 Extended Errors

The WebDAV client MAY use the errors that are returned by the server through the X-MSDAVEXT_ERROR header in order to get more information about server errors. Clients SHOULD NOT rely on these errors for anything other than information status.<17>

3.1.5.4 Adding Lock Headers to Existing Commands

The WebDAV Protocol locking semantics and processing orders, as specified in [RFC4918] section 8.10, are not altered by the WebDAV Protocol: Client Extensions. These extensions simply allow for fewer requests to be sent to the WebDAV server.

The LOCK headers SHOULD be added to the commands that read and write data from the WebDAV server when locking semantics are required. On read operations (GET or POST), the LOCK MAY be requested, refreshed, or unlocked. On write operations (PUT), the LOCK header that has the lock token SHOULD be added to successfully complete the operation.

The WebDAV client SHOULD add the LOCK extension headers only when it communicates with a server that supports these extensions.<18>

3.1.5.5 Retrying a Request with the 449 Status Code

If the client does not support retrying a request with the 449 status code, the client SHOULD treat the 449 Retry With status code as an unrecognized status code, as specified in [RFC2616] section 6.1.1, and SHOULD treat the Ms-Echo-Request header as an unrecognized entity header and ignore it, as specified in [RFC2616] section 7.1.

When a client that supports the 449 status code extension receives a 449 status code from the server, it MUST process the entity body of the response as normal but SHOULD NOT display its contents to the user.

After the entity body has been processed, the client MUST repeat the request that generated the 449 status code to the server and MUST include the Ms-Echo-Reply header with the new request.

3.1.6 Timer Events

No new timers are required except those in the WebDAV Protocol, as specified in [RFC4918].

3.1.7 Other Local Events

There are no new local events other than those that are specified in the WebDAV Protocol, as specified in [RFC4918].

3.2 WebDAV Server Details

3.2.1 Abstract Data Model

No new abstract data model is needed other than that described in the WebDAV Protocol, as specified in [RFC4918].

3.2.2 Timers

No new timers are required except those in the WebDAV Protocol, as specified in [RFC4918].

3.2.3 Initialization

No initialization is required except that in the WebDAV Protocol, as specified in [RFC4918].

3.2.4 Higher-Layer Triggered Events

No new events are triggered except those in the WebDAV Protocol, as specified in [RFC4918].

3.2.5 Processing Events and Sequencing Rules

A WebDAV server SHOULD advertise support for the WebDAV Protocol: Client Extensions by adding the X_MSDAVEXT: 1 header to the response to the OPTIONS command. If the server does not support the WebDAV Protocol: Client Extensions, the OPTIONS response MUST omit the X_MSDAVEXT header.

3.2.5.1 Translate Header

The WebDAV server MUST respond with the actual source of the file if the translate header exists with a flag value "f." The WebDAV server SHOULD perform access checks when it processes a command that contains a Translate: f header.<19>

To maintain consistency with Web browsers, the default behavior if this header is omitted is to translate the file; that is, omission of this header is the same as sending Translate: t.<20>

3.2.5.2 Adding Lock Headers to Existing Commands

The WebDAV server MUST honor lock headers that are sent on GET, POST, and PUT requests that are sent by using the Translate: F header if it advertises this functionality with X-MSDAVEXT: 1 in its OPTIONS response. It MAY honor these headers on requests that are not sent with Translate: F or on other methods. It SHOULD NOT send or honor these headers if it does not advertise this functionality with the X-MSDAVEXT: 1 OPTIONS response header.<21>

The following table provides the complete set of valid combinations that MAY be used for the WebDAV Protocol: Client Extensions. In the table, a value of Y indicates the presence of the extension while a value of N indicates omission of the extension. Any combination that is not included in this table MUST be processed as an HTTP 400 Bad Request error, as specified in [RFC2616] section 10.4.1.<22>

HTTP verb	Lock-token	Lock-time-out	Result
GET, HEAD or POST	Y	N	The header SHOULD be ignored.
GET, HEAD or POST	Y	Y	SHOULD fail the request if the token does not match. MUST fail the request if there is no existing lock that is associated with the specified token or if the file is locked by a different user. MUST refresh the lock if a nonzero time-out is specified. MUST unlock if timeout=0.
GET, HEAD or POST	N	Y	MUST fail the request if the file is already locked. MUST fail the request if timeout=0. MUST lock the file if no existing lock.
GET, HEAD or POST	N	N	MUST return the file without processing any locks.
PUT	Y	N	MUST process the PUT if the token matches. SHOULD fail the request if the token does not match or the file is not locked.
PUT	Y	Y	SHOULD fail the request if the token does not match. MUST fail the request if there is no existing lock that is associated with the specified token or the file is locked by a different user. MUST refresh the lock if a nonzero time-out is specified. MUST unlock if timeout=0.
PUT	N	Y	MUST fail the request if the file is already locked. MUST fail the request if timeout=0. MUST lock the file if no existing lock.
PUT	N	N	MUST process the PUT request, as specified in [RFC4918] section 9.7.

3.2.5.3 Extended Errors

The WebDAV server MAY add the extended error header in order to provide more error information to the client. The WebDAV server SHOULD NOT rely on the client to handle these errors because the WebDAV client MAY ignore the header.<23>

3.2.5.4 Extensions to GET and POST

The WebDAV server SHOULD ignore the header if it receives an X-MSDAVEXT: PROPFIND header when support for the WebDAV Protocol: Client Extensions is not advertised. If the WebDAV server advertised the support for the extensions, it MUST process the command and return the properties and the file contents in the format that is specified in section 2.2.5.<24>

3.2.5.5 Extensions to PUT

The WebDAV server MAY fail the request if it receives an X-MSDAVEXT: PROPPATCH header when the support for the WebDAV Protocol: Client Extensions is not advertised. If the WebDAV server advertised support for the extensions, it MUST process the command by retrieving the data from the body and processing both PROPPATCH (using the properties portion) and PUT. The command SHOULD be successfully completed only if both operations succeed.<25>

3.2.5.6 Retrying a Request with the 449 Status Code

If the server does not support retrying a request with the 449 status code, it MUST NOT send the 449 Retry With status code in an HTTP response message and SHOULD treat the Ms-Echo-Reply header as an unrecognized entity header and ignore it, as specified in [RFC2616] section 7.1.

A server implementing the 449 Reply With status code should behave as an HTTP server as specified in [RFC2616]. If the server needs the client to send more information before responding to the client's request for a specific resource, the server MAY return a 449 Retry With status code.

The 449 response MUST conform to the specification of an HTTP response message, as specified in [RFC2616] section 6.

When sending a 449 status code, the server MUST include the Ms-Echo-Request header.

If the server receives a request from the client that contains an Ms-Echo-Reply header, it MUST NOT reply to such a request with a 449 Reply With status.

3.2.6 Timer Events

No new timers are required except those in the WebDAV Protocol, as specified in [RFC4918].

3.2.7 Other Local Events

There are no new local events other than those that are described in the WebDAV Protocol, as specified in [RFC4918].

4 Protocol Examples

4.1 Translate Header

The following examples show the difference between requesting an entity and the source of an entity. This first example is a typical HTTP GET command as issued by a browser such as the Windows® Internet Explorer® browser.

Request:

```
GET /Temp/world.asp HTTP/1.1
Translate: t
Host: localhost
Accept: */*
```

Response:

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.1
Date: Wed, 28 Jun 2006 00:06:21 GMT
Content-Length: 129
Content-Type: text/html
Set-Cookie: ASPSESSIONIDCSSTSCQB=IEEJDPNAAIJECIOOBLMMGDJM; path=/
Cache-control: private
      <FONT SIZE="1">Hello World</FONT><BR>
      <FONT SIZE="2">Hello World</FONT><BR>
      <FONT SIZE="3">Hello World</FONT><BR>
```

An authoring application might want to retrieve the source of an entity, and then issue the same request, asking for the source of the entity, as shown in the following example.

Request:

```
GET /Temp/world.asp HTTP/1.1
Translate: f
Host: localhost
Accept: */*
```

Response:

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.1
Date: Wed, 28 Jun 2006 00:16:34 GMT
Content-Type: text/plain
Content-Length: 497
ETag: "22a87614489ac61:c02"
Last-Modified: Wed, 28 Jun 2006 00:16:19 GMT
Accept-Ranges: bytes
<%
'***********************************************
'*     Sample ASP Code                         *
'*                                              *
'************************************************
%>
<%
Dim I ' declare  loop variable
%>
<%
' Loop 3 times, adjusting the font size in each loop
For I = 1 To 3 Step 1
' Output our HTML and text using the value of I as
' the FONT TAG SIZE attribute.
%>
<FONT SIZE="<%= I %>">Hello World</FONT><BR>
<%
Next ' continue looping
%>
```

The only difference between these two requests is that the second request is requesting the source of an entity. This is a typical example of how the Translate header is used.

4.2 Extended Error Information

The following is an example of a response to a request to PUT a file that was checked out to another user.

```
HTTP/1.1 401 Unauthorized
Content-Length: 1656
Content-Type: text/html
X-MSDAVEXT_ERROR: 2342;
      The%20file%20is%20checked%20out%20to%20%22domain%5cusername%22
Server: Microsoft-IIS/6.0
WWW-Authenticate: NTLM
X-Powered-By: ASP.NET
Date: Tue, 25 Jan 2005 03:11:51 GMT
```

4.3 Example Command Combinations

The headers and extensions, as specified in section 2, enable multiple operations to be combined in a single request/response. The following list shows the typical combinations:

- POST or GET + PropFind
- POST or GET + Lock or Refresh or Unlock
- POST or GET + PropFind + Lock or Refresh or Unlock
- PUT + PropPatch
- PUT + Lock or Refresh or Unlock
- PUT + PropPatch + Lock or Refresh or Unlock

4.4 Example OPTIONS Command

The following is an example of an OPTIONS request and the corresponding response from a server that supports the WebDAV Protocol: Client Extensions.

Request:

```
OPTIONS / HTTP/1.1
translate: f
User-Agent: Microsoft-WebDAV-MiniRedir/5.2.3790
Host: office
Authorization: NTLM
Connection: Keep-Alive
Content-Length: 0
```

Response:

```
HTTP/1.1 200 OK
Date: Tue, 25 Jan 2005 03:12:26 GMT
Server: Microsoft-IIS/6.0
MicrosoftSharePointTeamServices: 12.0.2.6361
X-Powered-By: ASP.NET
MS-Author-Via: MS-FP/4.0,DAV
DAV: 1,2
Accept-Ranges: none
Allow: GET, POST, OPTIONS, HEAD, MKCOL, PUT, PROPFIND, PROPPATCH, DELETE, MOVE, COPY, GETLIB,
LOCK, UNLOCK
Cache-Control: private
Content-Length: 0
X-MSDAVEXT: 1
Public-Extension: http://schemas.fourthcoffee.com/repl-2
```

4.5 Example PUT + PROPPATCH + LOCK command

The following is an example of how the LOCK and PROPPATCH commands can be combined with a PUT command.

Request:

```
PUT /shared%20documents/Copy%20of%20Folder/test.rtf HTTP/1.1
translate: f
User-Agent: Microsoft-WebDAV-MiniRedir/5.2.3790
Host: dustinfrserver
Content-Length: 114234
Connection: Keep-Alive
X-MSDAVEXT: PROPPATCH
X-MSDAVEXTLockTimeout: Second-3600
Content-type: multipart/MSDAVEXTPrefixEncoded
Pragma: no-cache
Authorization: NTLM
<Properties-Size> <Properties> <File-Size> <File-contents>
```

Response:

```
HTTP/1.1 200 OK
Date: Tue, 30 Nov 2004 18:32:34 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
MicrosoftSharePointTeamServices: 6.0.2.5530
Cache-Control: private
Content-Length: 0
Public-Extension: http://schemas.fourthcoffee.com/repl-2
X-MSDAVEXTLockTimeout: Second-3600
Lock-Token: opaquelocktoken:{4A7A741A8}20041130T183232Z
```

4.6 Multipart Content Type

The following is an example of how the body of a request or response looks when the multipart content type is specified.

```
00000000000001BB<?xml version="1.0" encoding="utf-8" ?>
<D:propertyupdate xmlns:D="DAV:" xmlns:Z="urn:schemas-fourthcoffee-com:">
<D:set>
<D:prop>
<Z:Win32CreationTime>Wed, 20 Jun 2007 20:29:23 GMT</Z:Win32CreationTime>
<Z:Win32LastAccessTime>Wed, 20 Jun 2007 20:29:30 GMT</Z:Win32LastAccessTime>
<Z:Win32LastModifiedTime>Wed, 20 Jun 2007 20:29:30 GMT</Z:Win32LastModifiedTime>
<Z:Win32FileAttributes>00000020</Z:Win32FileAttributes>
</D:prop>
</D:set>
</D:propertyupdate>0000000000000013this is a text file
```

4.7 449 Response and Echo Reply

The following example shows a server returning a 449 response status along with a script to request that the browser set a cookie with the client's available screen dimensions. The client responds to the 449 response and echo request with an echo reply and the cookie that was generated as a result of executing the script.

Request:

```
GET /449/screen.449 HTTP/1.1
Host: localhost
Accept: */*
```

Response:

```
HTTP/1.1 449 Reply With
Content-Type: text/html
Ms-Echo-Request: token
Content-Length: 296

<html>
<head>
```

```
</head>
<body onload="createCookie();">
<script language="JavaScript">
<!--
function createCookie()
{
    strCookie = 'availWidth=' + screen.availWidth + '&availHeight=' + screen.availHeight;
    document.cookie = 'screen=' + strCookie;
}
--></script>
</body>
</html>
```

Echo reply:

```
GET /449/screen.449 HTTP/1.1
Host: localhost
Accept:*/*
Ms-Echo-Reply: token
Cookie: screen=availWidth=800&availHeight=600
```

Final response:

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 125

<html>
<head>
</head>
<body>
Please adjust your screen resolution to 1600 x 1200 to enjoy this content.
</body>
</html>
```

5 Security

5.1 Security Considerations for Implementers

WebDAV servers that support the translate: f header need to perform access checks before returning the source of the file, as specified in section 3.2.5.1, in order to protect any source content (for example, database passwords).<26>

5.1.1 Data Security Using File Encryption

WebDAV servers do not support encryption of files. WebDAV clients can use their own encryption mechanism and store the files in raw format. Files that are created by using the raw format are readable only from the WebDAV clients that know how to decrypt these files from the raw format.<27>

5.2 Index of Security Parameters

No security parameters are used in the WebDAV Protocol: Client Extensions.

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 Windows® 2000 operating system
- Windows® XP operating system
- Windows Server® 2003 operating system
- Windows Vista® operating system
- Windows Server® 2008 operating system
- Windows® 7 operating system
- Windows Server® 2008 R2 operating system

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.

<1> Section 2.1: Client support for SSL or TLS is available only in Windows Vista and Windows Server 2008. WebDAV servers that run on Internet Information Services, Windows SharePoint Services 2.0, or Windows SharePoint Services 3.0 support SSL/TLS.

<2> Section 2.1: Windows XP and Windows Server 2003 WebDAV clients only support port 80. Support for other ports is available only in Windows Vista and Windows Server 2008. The WebDAV client in Windows Vista and Windows Server 2008 uses port 80 by default for HTTP, and port 443 for HTTP over SSL or TLS. WebDAV servers that run on Internet Information Services, Windows SharePoint Services 3.0, or Windows SharePoint Services 2.0 support any port.

<3> Section 2.2.1: This header is supported by WebDAV clients only in Windows Vista and Windows Server 2008. This header is supported by the WebDAV server only in Windows SharePoint Services 3.0.

<4> Section 2.2.1: The WebDAV server in Windows SharePoint Services 3.0 recognizes this header only in the following instances:

- On an OPTIONS command, the WebDAV server sends the header in its OPTIONS response; it ignores it in the request.
- On a GET, HEAD, or POST request with a value of PROPFIND when used in conjunction with Translate: "f" | "F".
- On a PUT request with a value of PROPPATCH.

All other instances of this header are ignored.

<5> Section 2.2.2: The WebDAV server in Internet Information Services, Windows SharePoint Services 2.0, and Windows SharePoint Services 3.0 is more accepting of input for the Translate header. These implementations accept "f*" and "F*" as FALSE, where "*" is a wildcard, not a literal character. Everything else, including omission of the header, is accepted as TRUE.

<6> Section 2.2.2: This header is issued by the WebDAV client in Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008.

The WebDAV server in Internet Information Services applies the Translate header only to GET and HEAD requests. On Windows 2000 and Windows Server 2003, the WebDAV ISAPI uses the Translate header to determine whether request processing should be performed by the WebDAV ISAPI or passed on to the script mapped request handler. A value of FALSE will cause the WebDAV ISAPI to handle the request. Other values or a lack of a header will cause the WebDAV ISAPI to pass the request to the script mapped request handler. Handling of the PUT method is inconsistent without an explicit Translate header and may be handled either by the WebDAV ISAPI or passed to the script mapped request handler. The WebDAV ISAPI does not handle POST requests, so a POST request sent with a Translate value of FALSE will generate a 501 response.

Windows SharePoint Services 2.0 and Windows SharePoint Services 3.0 support this header on GET, POST, and HEAD commands, and all others ignore it.

<7> Section 2.2.3: Extended error handling is supported by the WebDAV client only in Windows Vista and Windows Server 2008; and by a WebDAV server that uses Windows SharePoint Services 3.0. If the WebDAV server includes this header in the response, the WebDAV client in Windows Vista and Windows Server 2008 propagates these errors to applications for their use. A complete list of errors that are supported is specified in section 3.1.5.3.

<8> Section 2.2.4: These headers are supported by a WebDAV client in Windows Vista and Windows Server 2008; and by a WebDAV server that is using Windows SharePoint Services 3.0 technology.

<9> Section 2.2.4.2: The WebDAV client in Windows Vista and Windows Server 2008 adds this header on requests that operate on a locked object against a WebDAV server that includes the X-MSDAVEXT: 1 header in its OPTIONS response. The WebDAV server that is implemented by Windows SharePoint Services 3.0 technology honors these headers on GET, POST, and HEAD requests if these requests also include the Translate: F header; and on PUT requests.

<10> Section 2.2.5: This extension is supported by the WebDAV client in Windows Vista and Windows Server 2008. This extension is supported by the WebDAV server in Windows SharePoint Services 3.0 technology.

<11> Section 3.1.1: The WebDAV client in Windows Vista and Windows Server 2008 keeps track of the server capabilities for the extensions and uses the new headers only against those servers that advertise this behavior.

<12> Section 3.1.5: The WebDAV client in Windows sends an OPTIONS command to verify server support for extensions. In Windows XP and Windows Server 2003, the WebDAV client does not support the extensions, and, therefore, the new headers are ignored on the OPTIONS response. In Windows Vista and Windows Server 2008, the WebDAV client does support the extensions, and the new headers in the server OPTIONS response are not ignored. Support for the HTTP 449 retry extension, as specified in section 3.1.5.5 and 3.2.5.6, is not indicated by the OPTIONS command. Clients may choose to support the HTTP 449 retry extension independent of their support for other WebDAV client extensions.

<13> Section 3.1.5.1.1: The WebDAV client in Windows Vista and Windows Server 2008 treats the response as an error if the WebDAV server advertised the support for the extensions and returns a

content type that is not multipart/MSDAVEXTPrefixEncoded in response to a request that has the X-MSDAVEXT: PROPFIND header.

<14> Section 3.1.5.1.1: These extensions are supported only by WebDAV clients in Windows Vista and Windows Server 2008. The WebDAV client adds this header only on GET and POST commands to the servers that advertised the support for extensions.

<15> Section 3.1.5.1.2: These extensions are supported only by WebDAV clients in Windows Vista and Windows Server 2008. The WebDAV client adds this header only on PUT commands to the servers that advertised the support for extensions.

<16> Section 3.1.5.2: The WebDAV client always adds the Translate header with Translate-value set to "f" on all commands that are sent to any WebDAV server.

<17> Section 3.1.5.3: In Windows XP and Windows Server 2003, the WebDAV client ignores the extended errors. In Windows Vista and Windows Server 2008, the WebDAV client uses selected extended errors, as specified in the following table. The client retrieves the numeric portion of the error and attempts to map it by using the following table. If the mapping is successful, the resulting Win32 error code is returned to the application. If the mapping is unsuccessful, the extended error information is returned as success, and only the string is available for the application.

Extended error	**Windows NT status code**	**Win32 error code (decimal)**	**Meaning**
0x0009000E	0xC0000901	220	V_DOC_CHECKED_OUT: The file is locked or checked out; therefore, the request failed.
0x00090075	0xC0000902	221	V_CHECKOUT_REQUIRED: The file must be checked out for the request to succeed.
0x0009006F	0xC0000903	222	V_BAD_FILETYPE_NO_URL: The server blocked the file because of its type.
0x0006000A	0xC0000904	223	V_SHTML_REQUEST_TOO_LONG: The request is too long.
0x000E0098	0xC0000905	224	V_FORMS_AUTH_NOT_BROWSER: The server is in forms-based authentication mode, and the client did not send authorization cookies.
0x00960004	0xC0000906	225	V_VIRUS_INFECTED_UL: The file was infected with a virus and cannot be uploaded.
0x00960009	0xC0000906	225	V_VIRUS_INFECTED_BLOCKED_DL: The file was infected with a virus and cannot be downloaded.
0x00960008	0xC0000907	226	V_VIRUS_DELETED_DL: The file was infected with a virus and was deleted because the virus already removed all the content.
0x00090070	0xC0000033	123	V_BAD_CHARS_IN_URL: The server does not support the URL.
0x00090071	0xC0000033	123	V_NO_RENAME_TO_THICKET_FOLDER: The server detected that the rename would have made a normal folder into a **thicket** supporting folder.

Extended error	Windows NT status code	Win32 error code (decimal)	Meaning
0x00090068	0xC0000106	206	V_URL_TOO_LONG: The URL was rejected because it was too long.
0x00090063	0xC0000801	1295	V_OVER_QUOTA: The change was rejected because the target site is over its disk quota.
UNKNOWN	0x00000000	0	

The string portion of the extended error is passed to the calling applications. The applications use this information to display the errors.

<18> Section 3.1.5.4: In Windows Vista and Windows Server 2008, the WebDAV client adds these headers to the requests when servers advertise support. These extensions and files are locked as part of normal file system locking semantics.

<19> Section 3.2.5.1: The WebDAV server in Internet Information Services requires that source access be specified for the file in its configuration in order to return the source of the file. On Windows 2000 and Windows Server 2003, file system write access to the file is also required.

Windows SharePoint Services 2.0 requires write access to the file in order to return the source of the file. Windows SharePoint Services 3.0 requires special access OpenItems in its configuration in order to return the source of the file.

<20> Section 3.2.5.1: The WebDAV server ISAPI extensions in Internet Information Services, Windows SharePoint Services 2.0, and Windows SharePoint Services 3.0, use the default header, translate: t.

<21> Section 3.2.5.2: Windows SharePoint Services 3.0 sends these headers in response to GET, POST, and HEAD requests against locked resources when the request also includes the Translate: F header. It honors these headers on PUT irrespective of the Translate: F header. Other Windows WebDAV servers (Windows SharePoint Services 2.0 and the Internet Information Services ISAPI WebDAV protocol implementation) do not honor or send these headers.

<22> Section 3.2.5.2: WebDAV servers in Internet Information Services and Windows SharePoint Services 2.0 do not support these headers. The WebDAV server in Windows SharePoint Services 3.0 supports this behavior but does not validate the lock token matches. It checks that the file is locked only when the client specifies a lock token and a time-out. As required, it also prevents PUT from succeeding and locks changes when the file is locked by a different user.

<23> Section 3.2.5.3: The Windows SharePoint Services 3.0 technology in Windows Server returns an error code for operations whenever it has additional information. Other Windows WebDAV protocol implementations do not return extended error information. Windows clients pass this information, when present, to the application as an extended error and perform mapping for known error codes, as specified in section 3.1.5.3.

<24> Section 3.2.5.4: WebDAV servers that receive an X-MSDAVEXT: PROPFIND header when the servers' support for the WebDAV Protocol: Client Extensions was not advertised ignore the X-MSDAVEXT request header and return only entity data without any properties.

<25> Section 3.2.5.5: Windows SharePoint Services 3.0 relies on the WebDAV client to send the Content-Type: multipart/MSDAVEXTPrefixEncoded header to get combined PUT and PROPPATCH behavior. Because clients are required to send this when they send the X-MSDAVEXT: PROPPATCH

header, it indirectly honors this header. Other WebDAV servers that run on Windows do not send the X-MSDAVEXT: 1 OPTIONS header and do not honor the X-MSDAVEXT: PROPPATCH header.

<26> Section 5.1: The WebDAV server in Internet Information Services requires that source access be specified for the file in its configuration in order to return the source of the file. On Windows 2000 and Windows Server 2003, file system write access to the file is also required.

Windows SharePoint Services 2.0 requires write access to return the source of the file. Windows SharePoint Services 3.0 technology requires special-access OpenItems in its configuration in order to return the source of the file.

<27> Section 5.1.1: The WebDAV client supports encryption of files that are stored on Internet servers. Encryption of files can be controlled by the application using the specific encryption APIs that are described with the file system reference. For more information, see [MSDN-STC]. Optionally, users can choose what files to encrypt by using Internet Explorer to access and modify the advanced properties of the file.

The WebDAV client uses the Encrypting File System (EFS) of the NTFS file system to encrypt files that are stored on WebDAV servers. The server does not see the plaintext version of the file, nor is it provided with any encryption key by which it can access the file.

When users use file encryption, to ensure that sensitive data is kept secure, it is critical that they take proper precautions and save a copy of their certificates in a secure place.

The WebDAV client uses the standard GET and PUT commands to get the binary data over the wire and then decrypts it locally by using the NTFS Encrypting File System.

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index