[MS-FPSE]:
FrontPage Server Extensions Remote Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
12/18/2006	0.1		MCPP Milestone 2 Initial Availability
03/02/2007	1.0		MCPP Milestone 2
04/03/2007	1.1		Monthly release
05/11/2007	1.2		Monthly release

Date	Revision History	Revision Class	Comments
06/01/2007	1.2.1	Editorial	Revised and edited the technical content.
07/03/2007	2.0	Major	Section 2.3 Added information about how and when an extra 401 response can be sent from the server.
07/20/2007	2.0.1	Editorial	Revised and edited the technical content.
08/10/2007	2.0.2	Editorial	Revised and edited the technical content.
09/28/2007	3.0	Major	Updated and revised the technical content.
10/23/2007	3.0.1	Editorial	Revised and edited the technical content.
11/30/2007	4.0	Major	Added and revised sections.
01/25/2008	5.0	Major	Updated and revised the technical content.

Table of Contents

1 Introduction

The FrontPage Server Extensions Remote Protocol specifies a set of server extensions that can be used to augment a basic HTTP server. These extensions provide file server functionality similar to WebDAV, allowing a Web site to be presented as a file share. The use of WebDAV is recommended over the FrontPage Server Extensions Remote Protocol. For more information about WebDAV, see [MS-WDV].

The FrontPage Server Extensions Remote Protocol uses HTTP version 1.1, as specified in [RFC2616], as a transport. Requests are specialized form posts, and responses are in HTML, as specified in [RFC2854]. Despite the use of HTTP, the protocol is intended to be used by a client program, not by the user directly through a Web browser.

The FrontPage Server Extensions Remote Protocol is a subset of a larger protocol known as FrontPage Server Remote Protocol Extensions. The FrontPage Server Extensions Remote Protocol is the protocol that is used when communicating between Windows clients and Windows servers. The larger protocol is used to perform a wider array of Web site administration. Because all implementations of the FrontPage Server Extensions Remote Protocol on Windows also implement FrontPage Server Remote Protocol Extensions , be aware that Windows servers may not ignore arguments and methods noted in this document as messages that a client must not send.

The SharePoint Team Services dialogview is an application of the FrontPage Server Extensions Remote Protocol that is addressed in this document because it has certain behaviors apart from the normal FrontPage Server Extensions Remote Protocol communications. The purpose of the dialogview is to allow a client to display a server-rendered HTML-based rendering of the files located on a particular **Web site**.<1>

1.1 Glossary

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

Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
Unicode
Uniform Resource Locator (URL)

The following terms are specific to this document:

Bot: FrontPage objects that are evaluated and executed when an author saves a page or, in some cases, when a site visitor browses to a page.

Default Web Page: All folders on a **Web site** have a **default Web page** (also called a home page). This page is displayed when a user browses to that folder. The **default Web page** is usually Index.htm or Default.htm, although it can be any page the site creator or the user specifies.

Derived Documents Folder: Server implementations may need to generate temporary intermediate files or files not directly uploaded by the client. These files are commonly stored in a folder whose **service-relative** URL is _derived. On Microsoft Windows implementations, it includes *.htx files created by the FrontPage Search component and composite text on .gif images, such as those used for **theme elements**. The contents of the folder are considered a server implementation detail, but the list documents method, as specified in section 3.1.5.3.8, allows the client to ask the server to filter out the contents of this folder.

Dictionary: A collection of pairs of items. Each pair consists of a key, which is a string and a value that can be of any type. Items in the **dictionary** are retrieved by providing a key for which the **dictionary** returns the associated value.

Document Checkout: A method used to lock a document to prevent users from concurrently editing the document. For details on **document checkout**, see section 3.

Document Library: A List created as a container for Documents. A Document Library stores Documents and Folders. A Document Library can support publishing status values for Documents (e.g. Draft, Checked-In, Checked-Out, and Published).

Executable Folder: Web servers commonly support the notion of a server-side executable in which pages are rendered by running a routine rather than by returning static contents from a file. These servers generally allow the administrator to enable and disable this feature on a folder-by-folder basis. A folder is called an **executable folder** if this feature is enabled for files within it.

Folder: A container within a List that acts like a file system directory. A Folder can contain other Folders, Documents, or List Items.

Form: A Page that allows the creation, viewing or editing of List Items in a List.

Hidden Documents: Files and folders whose URLs contain a path component that begins with an underscore (_).

Link Fixup: Some document formats support the notion of a link where one document references another document. The server can discover these links to referenced documents and rewrite the links as documents are moved or copied, so the links do not go to stale references. As the server rewrites these links, it is said to be doing **link fixup**.

Long-Term Checkout: A **document checkout** that rejects edits against the file by other clients but, unlike a **short-term checkout**, does not expire unless the client application sends the uncheckout document request, as specified in section 3.1.5.3.15. FrontPage Server Extensions Remote Protocol–compliant clients never use **long-term checkout**, and servers may choose to ignore this value.

Metadata: Data that describes an object. Metadata may be associated with Site Collections, Sites, Documents and other objects.

Metadict: A **dictionary** with strongly typed values.

Metakey: The string used to look up a value in a **metadict**.

Nesting Level: A count used during the formatting of messages. It is the number of times an open-bracket (OBRACKET) is sent minus the number of times a close-bracket (CBRACKET) is sent. For details, see section 2.2.1.1.3.

Page: A Document consisting of HTML that may contain dynamic content such as Web Parts that are interpreted before display to a client application.

Server-Relative: A **server-relative** URL defines the location of an item in relation to the root of the server. A server's URL can be determined by using the url to web url method, as specified in section 3.1.5.3.16.

Service-Relative: A **service-relative** URL defines the location of an item in relation to the root of the Web. For example, if a page is located in the root folder of a Web, the **service-relative** URL consists simply of the page name. The FrontPage Server Extensions Remote Protocol

often returns **service-relative** URLs that the client must interpret. A service's URL can be determined by using the url to web url method, as specified in 3.1.5.3.16.

Shared Borders Folder: FrontPage implements a feature in which HTML pages may be authored to contain common elements. These elements are called shared borders, and their HTML contents are stored in a directory whose **service-relative** URL is _borders. This folder is known as the **shared borders folder**.

Short-Term Checkout: A **document checkout** that automatically expires after a set period of time. While a client application has a **short-term checkout** of a file, edits against the file by other clients are rejected. If the client application does not renew the **short-term checkout**, edits by other clients are allowed after the **short-term checkout** expires.

Short File Name: A file name that consists of up to eight characters, a period, and up to three characters.

Site: A URL-based container within a Site Collection that may contain Documents, Document Libraries, Lists, and child Sites known as SubSites. The structure and content of Sites, when created, are based on implementation-specific Site Templates such as, but not limited to, Team Sites, Document Workspaces, and Meeting Workspaces. Also referred to as Web Site.

Subsite: A Site whose URL is composed in part by the URL of a parent Site within the same Site Collection. A Subsite parent may be either the Root Site of the Site Collection or another Subsite. Each subsite can have independent administration, authoring, and browsing permissions from the root Web site and other subsites.

Subweb: A subordinate Web, that is, the named subdirectory of the root **Web site** that is itself a complete FrontPage-based **Web site**. Each **subweb** can have independent administration, authoring, and browsing permissions from the root **Web site** and other **subwebs**.

Task-List Files: The FrontPage client has a task list feature that stores information in two files on the server. The **task-list files** are always stored in _vti_pvt/_x_todo.htm and _vti_pvt/x_todoh.htm.

Theme Elements: A set of coordinated graphic elements applied to a document or Web page or across all pages in a **Web site**. Themes can consist of designs and color schemes for fonts, link bars, and other page elements.

Thicket: A group of supporting files and folders on the Web server that together store the contents of one logical document, for example, a Web page or Microsoft Word document that includes pictures.

Web Site: An autonomous service. A server's URL namespace is decomposed into a number of **Web sites**, each with a contiguous namespace. Each **Web site** has its own entry points, metadata, and independent administration, authoring, and browsing permissions.

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

[MS-DTYP] Microsoft Corporation, "Windows Data Types", January 2007.

[RFC1123] Braden, R., "Requirements for Internet Hosts–Application and Support", RFC 1123, October 1989, http://www.ietf.org/rfc/rfc1123.txt

[RFC1341] Borenstein, N., and Freed, N., "MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1341, June 1992, http://www.ietf.org/rfc/rfc1341.txt

[RFC1866] Berners-Lee, T. and Connolly, D., "Hypertext Markup Language - 2.0", RFC 1866, November 1995, http://www.ietf.org/rfc/rfc1866.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

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

[RFC2854] Connolly, D. and Masinter, L., "The 'text/html' Media Type", RFC 2854, June 2000, http://www.ietf.org/rfc/rfc2854.txt

[RFC4234] Crocker, D., Ed. and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005, http://www.ietf.org/rfc/rfc4234.txt

1.2.2 Informative References

[MS-WDV] Microsoft Corporation, "Web Distributed Authoring and Versioning (WebDAV) Protocol: Client Extensions", August 2007.

1.3 Protocol Overview (Synopsis)

The FrontPage Server Extensions Remote Protocol is used by client applications to display the contents of a Web site as a file system. The FrontPage Server Extensions Remote Protocol provides file uploading and downloading, directory creation and listing, basic file locking, and file movement on a Web server by using a set of methods.

Each method is an HTML form POST, as specified in [RFC2854], that accepts a set of parameters and returns a set of values as an HTML response. The method parameter defines what operation the server will perform in addition to the meanings of the other parameters and return values.

The client sends method call requests to the server, and the server sends return values to the client via HTML. The server never initiates any communication with the client. All communication is transported over **HTTP** or secure HTTP (**HTTPS**), as specified in [RFC2616] section 9.1. Method calls are sent as HTTP posts with arguments as post arguments, and server responses are sent as a list in the body of an HTTP response. All posts are made to one of several well-defined **URLs** on the server, which can be discovered by clients.

The following sequence diagram depicts a generic Microsoft FrontPage Server Extensions conversation. A brief explanation of each message follows, and details are defined in sections 2 and 3.

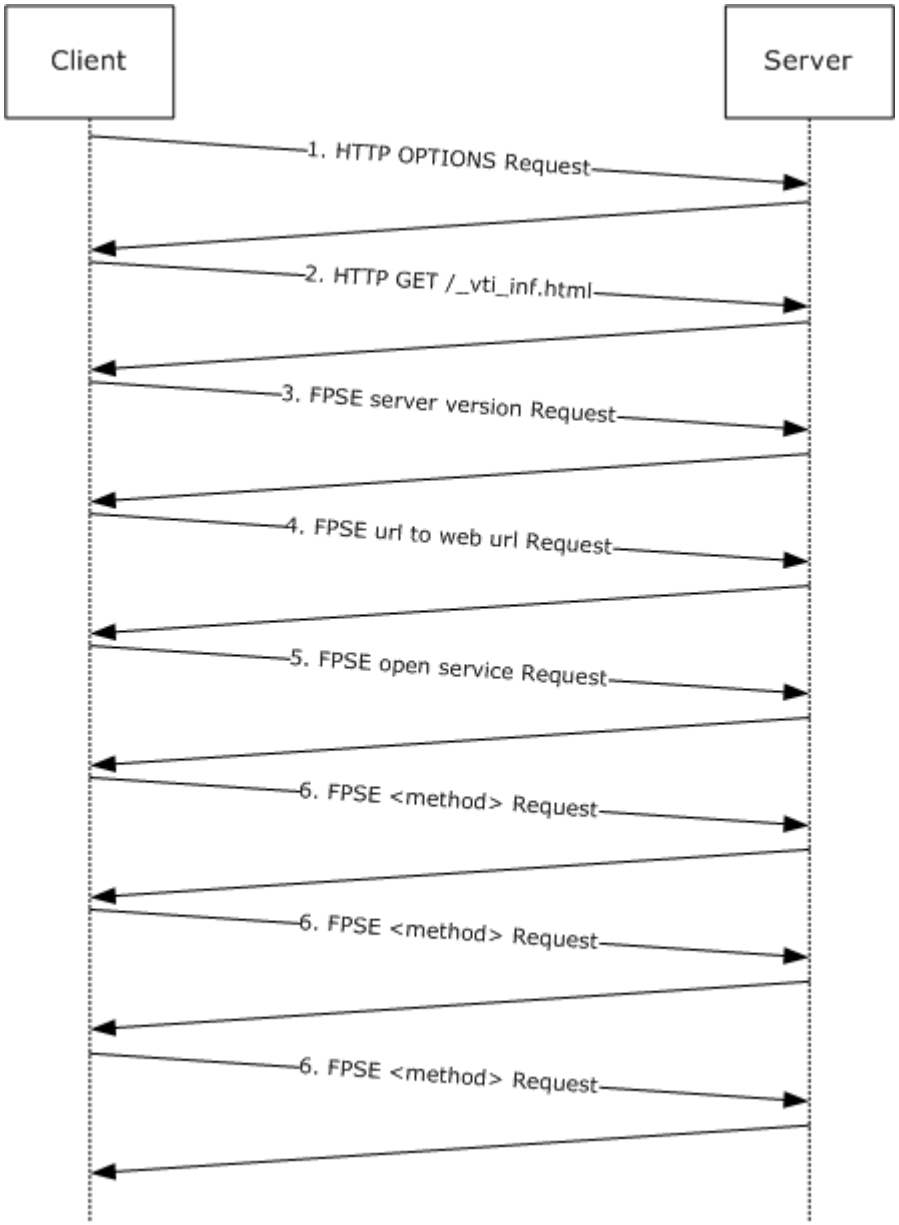

Figure 1: A generic Microsoft FrontPage Server Extension message sequence

1. The HTTP OPTIONS request is sent to determine if the server supports the FrontPage Server Extensions Remote Protocol. If the response contains the MS-Author-Via header, as specified in section 3.1.3.1, the server supports the protocol. This value is often cached by clients.

2. The HTTP GET on _vti_inf.html will return information specifying the well-defined URLs to which the client must POST further method calls.

3. At this point, the client is prepared to start making method calls against the server. The first call is a server version request (section 3.1.5.3.14) whereby the client negotiates a protocol version with the server.

4. The client MAY then call url to web url (section 3.1.5.3.16) if the Web site is a **subweb**, that is, not located at the root of the server's namespace.

5. Next, the client MAY make an open service request(section 3.1.5.3.10)on the Web it wants to open. This request is optional, but it will return information about the Web site's capabilities, such as support for version control.

6. The client MAY make any method calls against the server. The nature of any further client/server communication is determined by the specific needs of the client at the time.

To give an example of how the protocol is used in Windows, a user might type an HTTP URL into the address bar of a file browser. The file browser contacts the server to determine if it supports the FrontPage Server Extensions Remote Protocol. If so, the file browser uses the FrontPage Server Extensions Remote Protocol to list the contents of the directory that the user entered and to display the contents just as it would display files on the local hard disk.

The SharePoint Team Services dialogview aspect of the FrontPage Server Extensions Remote Protocol is designed to allow a client to ask the server to provide an HTML rendering of its file structure. The client can then display this HTML rendering and navigate through it.

When using SharePoint Team Services dialogview, section 3.1.5.3.17.1, the sequence of messages is similar to that shown in Figure 1. For a brief explanation of each message, see sections 2 and 3.

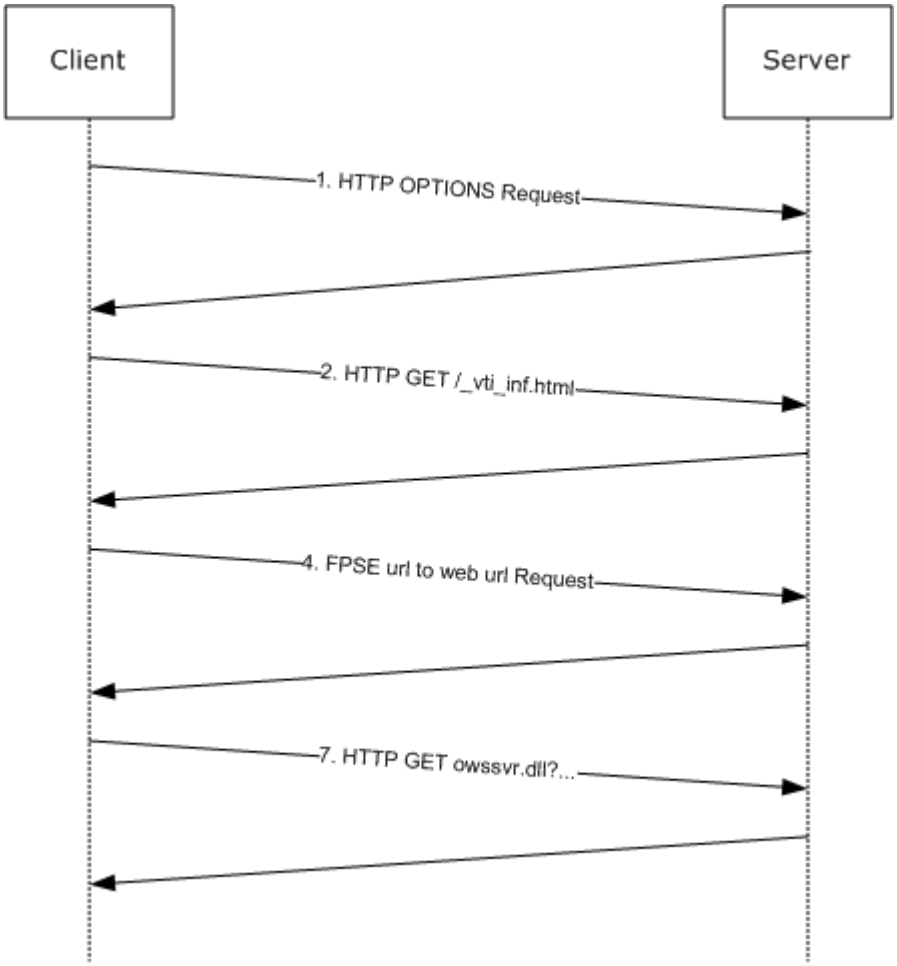

Figure 2: SharePoint Team Services dialogview message sequence

The first three messages are used identically to those shown in Figure 1. The final HTTP GET returns the HTML that the client SHOULD use to display the Web site's file system. This GET must be performed against a specific, well-known URL on the server (see section 3.1.3.2).

1.4 Relationship to Other Protocols

The FrontPage Server Extensions Remote Protocol is transported via HTTP version 1.1 POSTs and responses, as specified in [RFC2616] sections 9.5 and 6, respectively.

SharePoint Team Services dialogview (section 3.1.5.3.17.1) relies on the FrontPage Server Extensions Remote Protocol to discover the name and location of the Microsoft Office Web Services Server executable by using the information returned in _vti_inf.html. Windows extensions to WebDAV use its error codes, as specified in [MS-WDV].

1.5 Prerequisites/Preconditions

The client must know the URL of the server it wants to communicate with, which is usually passed by the user as the prompt for beginning the FrontPage Server Extensions Remote Protocol conversation. If required by the server, the client must authenticate by using the underlying HTTP mechanisms, as specified in [RFC2616] section 14.8.

1.6 Applicability Statement

The FrontPage Server Extensions Remote Protocol is a precursor to the WebDAV protocol and may be used in similar situations. Because it is an earlier technology, most developers will find WebDAV, as specified in [MS-WDV], a more appealing option.

The SharePoint Team Services dialogview aspect, as specified in section 3.1.5.3.17.1, of the protocol may be used by a client application to allow the server to control the appearance of a file store to a user. For instance, if the server has richer metadata semantics than the typical client can display or uses a non-standard file and folder hierarchy, this view can be used to offload rendering of the file navigation to the server.

1.7 Versioning and Capability Negotiation

1.7.1 Protocol Versions

Version negotiation is performed by using the server version method, section 3.1.5.3.14. The client sends its own protocol version in the method name section of the request. The server compares that to the server protocol version and replies to the client. The protocol version the server uses is given in the response header in the form of (Min(ServerVersion, ClientVersion)). The client is expected to use this version for any remaining communications. If the version of the client or server is not supported, the one with the newer protocol version discontinues the conversation.

The structure of a FrontPage Server Extensions Remote Protocol version (see section 2.2.2.5.1), as defined, has four parts: a major version, minor version, phase number, and build number. Thus, a version might look like 1.0.0.3214. Versions grow over time, so 3.0 is considered earlier, or older, than 4.0.

In the FrontPage Server Extensions Remote Protocol, the client, server, and protocol each have their own version, although all of them follow the same format. The client and server version are used in the negotiation to determine the protocol version. For details, see section 3.1.5.3.14.

All servers reject any client with a version that is earlier than 4.0.2.2611, and clients reject any server with a version that is earlier than 3.0.2.1002. The server returns a V_RPC_CLIENT_TOO_OLD (0x0004000C) error code if an incompatible client is encountered. If the version of the server is not supported, the client simply ignores the server, and no further communication with the server is attempted.

The FrontPage Server Extensions Remote Protocol components were shipped as part of Windows and as separate Web downloads. Only the major and minor version numbers are listed. Except as noted above, any numbers that meet the requirements (see 2.2.2.5.1) may be used for the phase and build numbers. The following table specifies which version of the FrontPage Server Extensions Remote Protocol is contained in each version of Windows.

Revision Summary	In-Box FPSE Server	Other Supported FPSE Servers
Windows NT 3.51 and earlier	None	None
Windows NT 4.0	None	None
Windows XP (x86)	4.0	None
Windows XP 64-Bit Edition	None	None
Windows Server 2003 (x86)	5.0	5.0, 6.0, 12.0
Windows Server 2003 (x64)	None	6.0, 12.0
Windows Server 2003 R2	Microsoft FrontPage Server Extensions 5.0, Windows SharePoint Services 2.0	5.0, 6.0, 12.0
Windows Vista	None	None
Windows Server 2008	12.0	5.0

For the remainder of this document, the protocol version will be referred to instead of the Windows version to describe support for protocol functionality.

1.7.2 Capability Negotiation

The Microsoft FrontPage Server Extensions clients and servers perform capability negotiation because some operations are only supported by newer servers. This negotiation is performed by using the Web site meta information that is returned in the server version method, as specified in section 3.1.5.3.14. The meta information contains a set of keys and values. Clients can determine server capabilities by looking for certain keys in the meta information. Information regarding **metakeys** that denotes specific behaviors the server may or may not support are detailed in section 2.

1.8 Vendor-Extensible Fields

There are no vendor-extensible fields in the FrontPage Server Extensions Remote Protocol.

1.9 Standards Assignments

The FrontPage Server Extensions Remote Protocol does not use any standards assignments other than those of HTTP 1.1, as specified in [RFC2616].

2 Messages

The following sections specify the message transport, message syntax, data types, and messages that are used in the FrontPage Server Extensions Remote Protocol.

2.1 Transport

The FrontPage Server Extensions Remote Protocol uses HTTP version 1.1, as specified in [RFC2616], as transport for the GET and POST methods.

2.1.1 Client Requests

Client requests to the server MUST be transmitted as GET methods appended to a URL, hereafter referred to as the URL Mode. For details about the syntax, see section 2.2.1.

If the client request does not conform to the message definitions that follow, the server MUST return a syntax error to the client and stop parsing the request.

2.1.2 Server Responses

Server responses to client requests MUST be transmitted as POST responses in HTML, as specified in ([RFC2854]), and are hereafter referred to as HTML Mode. Exceptions are if the server responses are otherwise specified. For details about the syntax, see section 2.2.1.

If the server response does not conform to the message definitions that follow, the client MUST ignore the server response and stop communication with the server.

2.2 Message Syntax

This section specifies the Microsoft FrontPage Server Extensions syntax and the data types that are used when a Windows client posts FrontPage Server Extensions Remote Protocol requests to a server. It also specifies the syntax that is used by the server to respond to client requests. The syntax and data types are defined using Augmented Backus–Naur Form (ABNF), as specified in [RFC4234].

2.2.1 Syntax

The FrontPage Server Extensions Remote Protocol is used in URL Mode and HTML Mode in client request and server responses, respectively. These two modes differ with respect to encoding rules and the values of certain tokens in the stream. Implementations MUST use the following syntax rules that define these encoding schemes.

All FrontPage Server Extensions Remote Protocol communications are case sensitive. The reader should assume that all strings are case sensitive unless otherwise noted.

2.2.1.1 Syntax Delimiters

The following two sections specify primitives that are used as punctuation within strings in the full syntax for both URL Mode and HTML Mode, respectively. They are defined for both URL Mode and HTML Mode, so that in the remainder of the document a single definition can be given for higher-level constructs.

2.2.1.1.1 URL Mode

The listed primitives are used as punctuation within a string in URL Mode.

```
PARGSEP = "&"
SARGSEP = ";"
PVALSEP = "="
SVALSEP = "="
LISTSEP = ";"
OBRACKET = "["
CBRACKET = "]"
STARTLIST = ""
```

2.2.1.1.2 HTML Mode

The listed primitives are used as punctuation within a string in HTML Mode.

```
PARGSEP = LF
SARGSEP = LF "<li>"
PVALSEP = "="
SVALSEP = "="
LISTSEP = LF "<li>"
OBRACKET = LF "<ul>"
CBRACKET = LF "<ul>"
STARTLIST = LF "<li>"
```

2.2.1.1.3 Nesting Level Dependent Elements

An implementation of the FrontPage Server Extensions Remote Protocol MUST keep track of the number of times an OBRACKET is sent minus the number of times a CBRACKET is sent in the current request. Hereafter, the value will be referred to as the **nesting level**. This value affects which delimiters MUST be used.

If the nesting level is zero:

```
ARGSEP = PARGSEP
VALSEP = PVALSEP
```

Otherwise, if the nesting level is not zero,

```
ARGSEP = SARGSEP
VALSEP = SVALSEP
```

2.2.1.2 Character Escaping

The FrontPage Server Extensions Remote Protocol uses UTF-8, as specified in [MS-DTYP], as its character encoding. In every instance that follows in this document in which a string is referred to as a literal, it may be assumed that the character is UTF-8 encoded. Depending on the mode, URL Mode or HTML Mode, various character escaping is used, as shown in the following two sections.

2.2.1.2.1 URL Mode

In URL Mode, characters are escaped as follows:

```
ESCAPED-CHAR = ALPHA / DIGIT ; literal meaning
```

```
/ "+" ; encoded space
/ "%5c%5c" ; encoded backslash
/ "%5c%5c" ; encoded backslash
/ "%5c%3d" ; encoded equal sign
/ "%5c%5b" ; encoded open bracket
/ "%5c%5d" ; encoded close bracket
/ "%5c%3b" ; encoded semicolon
/ "%5c%22" ; encoded double quote
/ "%" 2HEXDIG ; anything not mentioned above
```

A sender SHOULD encode in this order (for example, a space SHOULD be encoded as "+" rather than "%20"; an A SHOULD be encoded as "A" rather than "%41"). A receiver MUST decode "%" 2HEXDIG and + to a space. A backslash MUST be ignored except:

1. If it is followed by another backslash, in which case it must be treated as a backslash.
2. If it is followed by =, [,], or ;, the backslash is ignored but the character that comes after MUST not be treated as a delimiter.

2.2.1.2.2 HTML Mode

In HTML Mode, characters are escaped as follows:

```
ESCAPED-CHAR =
%d32-33 / %d35-58 / %d63-91 ; literal meaning
/ %d93-122 / %d124 / %d126-127 ; literal meaning
/ "\t" ; encoded tab (%d8)
/ "\b" ; encoded backspace (%d9)
/ "\n" ; encoded newline (%d10)
/ "\f" ; encoded formfeed (%d12)
/ "\r" ; encoded carriage return (%d13)
/ "&#" 2DIGIT ";" encoded non-printing characters that are not
                  specially handled (%d0-7 / %d11 / %d14-31) or
                  special printing characters (%d34 / %d59-62 / %d92)
/ "&#" 3DIGIT ";" special printing characters (%d123 / %d125) or
                  non-printing 3 digit characters (%d128-255)
```

That is, send "\t" (3rd expansion) rather than "&#08;" (8th expansion), and send "<" (8th expansion) rather than "<" (9th expansion). However, a receiver MUST accept any of these forms.

2.2.2 Data Types

This section describes the data types that are used when the Windows client posts FrontPage Server Extensions Remote Protocol requests to the server, and the server responds to the client.

2.2.2.1 Primitive Data Types

This section specifies the primitive data types, as specified in [RFC4234], that are used in the FrontPage Server Extensions Remote Protocol.

2.2.2.1.1 UNSIGNED-INT

The UNSIGNED-INT data type is defined in the following code sample.

```
UNSIGNED-INT = 1*DIGIT ; default value = "0"
```

2.2.2.1.2 INT

The INT data type is defined in the following code sample.

```
INT = 1*UNSIGNED-INT / "-" UNSIGNED-INT
```

2.2.2.1.3 BOOLEAN

The BOOLEAN data type is defined in the following code sample.

```
BOOLEAN = "true" / "false" ; default value = "false"
```

2.2.2.1.4 DOUBLE

The BOOLEAN data type is defined in the following code sample.

```
DOUBLE = INT [ "." UNSIGNED INT ] / "." UNSIGNED-INT
```

2.2.2.1.5 STRING

The STRING data type is defined in the following code sample.

```
STRING = *ESCAPED-CHAR
```

Note STRING represents a **Unicode** string with each ESCAPED-CHAR corresponding to a byte in a UTF-8 sequence. For instance, the "æ" character (a combined "ae") is "U+00e6", which has a UTF-8 representation of "%xc3.a6". Thus, the string "Cæsar" can be represented as "C%c3%a6sar" in URL Mode and as "CÃ¦sar" in HTML Mode.

2.2.2.1.6 TIME

The TIME data type is defined in the following code sample.

```
TIME = STRING
```

Note TIME values MUST conform to the Greenwich Mean Time (GMT) format, as specified in [RFC1123] section 5.2.14.

2.2.2.2 Shared Elements

This section specifies some shared elements that are used in both requests and responses.

```
RPCKEY = [ARGSEP] RPCKEY-KEY-STRING VALSEP
```

The leading ARGSEP MUST be present, except in URL Mode, if it is the first key after an OBRACKET or at the start of a response, in which case it MUST NOT be present.

```
RPCKEY-KEY-STRING = STRING
RPCVALUE = UNSIGNED-INT / INT / BOOLEAN / DOUBLE / STRING / TIME /
VERSION / VECTOR-X / DICT / METADICT / DOCINFO / DOCUMENT-RETURN-TYPE /
DOCUMENT-LIST-RETURN-TYPE / SERVICE-RETURN-TYPE / DOC-INFO-REQUEST /
URL-DIRECTORY / STATUS / PUT-OPTION / RENAME-OPTION
URL-STRING = STRING
```

The string MUST be a correctly formatted URL.

```
PROTOCOL-VERSION-STRING = UNSIGNED-INT "." UNSIGNED-INT "."
UNSIGNED-INT "." UNSIGNED-INT
METHOD-KEY-VALUE = RPCKEY RPCVALUE
```

The RPC-KEY-STRING in the RPCKEY of a METHOD-KEY-VALUE MUST be "method", and the RPCVALUE must be an encoded METHOD-VALUE defined as:

```
METHOD-VALUE = REQUEST-NAME-STRING [":" PROTOCOL-VERSION-STRING]
```

The REQUEST-NAME-STRING MUST be one of the values defined in section 2.2.2.1. If a client knows the server version, it SHOULD send the protocol version as the minimum of the client version and server version; otherwise, the client SHOULD send the client version. In its responses, the server MUST set the protocol version as the client's requested protocol version and the server version.

2.2.2.3 Request Syntax

This section specifies the syntax for a FrontPage Server Extensions Remote Protocol request. For details on which arguments should be sent for each method, refer to section 3.1.5.3.1.

```
REQUEST = METHOD-KEY-VALUE *(ARG-NAME ARG-VALUE) LF
```

The arguments for the request are the set of ARG-NAME ARG-VALUE elements that appear after the METHOD-KEY-VALUE.

```
ARG-NAME = RPCKEY
ARG-VALUE = RPCVALUE
```

The elements METHOD-KEY-VALUE, RPCKEY, and RPCVALUE are defined as specified in section 2.2.2.2.

2.2.2.4 Response Syntax

This section specifies the syntax for a FrontPage Server Extensions Remote Protocol response. The specifics of which return values should be sent for each method are given in section 3.1.5.3.

```
RESPONSE = "<html><head><title>Vermeer RPC packet</title><head>" LF
"<body>" METHOD-KEY-VALUE *(RET-NAME RET-VALUE)"</body>" LF "</html>"
LF
```

The return values are the set of RET-NAME RET-VALUE elements that appear after the REQUEST-NAME-STRING.

```
RET-NAME = RPCKEY
RET-VALUE = RPCVALUE
```

The elements METHOD-KEY-VALUE, RPCKEY, and RPCVALUE are defined in section 2.2.2.2.

2.2.2.5 Complex Data Types

This section specifies the complex data types that are used in method requests and responses. These values, in addition to the primitive data types, will be used throughout section 3.1.5.3 to define the data types for arguments and return values.

2.2.2.5.1 Version

Used to communicate a version number. The default value is "0.0.0.0".

```
VERSION = OBRACKET "major ver" VALSEP INT ARGSEP "minor ver" VALSEP
INT ARGSEP "phase ver" VALSEP INT ARGSEP "ver incr" VALSEP INT
CBRACKET
```

Major and minor versions are 1.0, 1.1, 2.0, 3.0, 4.0, 5.0, 6.0, and 12.0. In phase version, the values are 0, 1, 2, or 3. The number 0 represents an alpha release or earlier; 1 represents a beta release; 2 represents an official release; 3 represents an updated version increment that is used to differentiate, for example, SP1 from SP2, or internal builds before release.

Note Version numbers are ordered numerically, not lexicographically. For example, 12.9 is earlier than 12.10.

2.2.2.5.2 Vector

A vector is a typed array of elements the default value of which is empty.

```
VECTOR-X = OBRACKET STARTLIST 1*(X LISTSEP) CBRACKET
```

All data types can have a vector type associated with them where X, as in the example above, represents the vector data type. For instance, VECTOR-STRING = OBRACKET STARTLIST 1*(STRING

LISTSEP) CBRACKET. X can be a simple type, such as STRING, or a complex type, such as DOCINFO.

2.2.2.5.3 Dictionary

The FrontPage Server Extensions Remote Protocol form of a **dictionary**. MetaDictionaries are just like normal dictionaries, except the value contains data type and access information.

```
KEY-STRING = STRING
```

The key that is used to look up the value.

```
VALUE-STRING = STRING
```

The value that is found with the key.

```
DICT = OBRACKET [STARTLIST KEY-STRING LISTSEP VALUE-STRING *(LISTSEP
KEY-STRING LISTSEP VALUE-STRING)] CBRACKET ; default value = empty

METADICT = DICT ; default value = empty
```

2.2.2.5.4 MetaDictionary

For **METADICTs**, the VALUE-STRING, when decoded, MUST be in a special form represented as METADICT-VALUE.

```
METADICT-VALUE = "T" METADICT-CONSTRAINT-CHAR "|" TIME
/ "V" METADICT-CONSTRAINT-CHAR "|" METADICT-STRING-VECTOR
/ "B" METADICT-CONSTRAINT-CHAR "|" BOOLEAN
/ "U" METADICT-CONSTRAINT-CHAR "|" METADICT-INT-VECTOR
/ "D" METADICT-CONSTRAINT-CHAR "|" DOUBLE
/ "S" METADICT-CONSTRAINT-CHAR "|" STRING

METADICT-CONSTRAINT-CHAR = "X" / "R" / "W"
X: The client MUST ignore the value.
R: The client MAY read the value but MUST NOT write the value.
W: The client MAY read or write the value.

METADICT-INT-VECTOR = / METADICT-INT-VECTOR SP INT

METADICT-STRING-VECTOR = / METADICT-STRING-VECTOR SP
                           METADICT-STRING-ITEM

METADICT-STRING-ITEM = *METADICT-STRING-ITEM-CHAR

METADICT-STRING-ITEM-CHAR = %x1-1F / %x21-5b / %x5d-ff ; unescaped
/ %x5c SP; escaped space
/ %x5c %x5c; escaped backslash
```

2.2.2.5.5 DocInfo

Contains a document name and its metadata.

```
DOCINFO = OBRACKET "document_name" VALSEP URL-STRING ARGSEP
"meta_info" VALSEP METADICT CBRACKET
```

A DOCINFO assumes the URL specified by the *document_name* parameter is **service-relative**.

Example (encoded as sent over the wire):

```
%5bdocument%5fname%3dfolder1%2ffolder2%2fsmall%2etxt%3bmeta%5finfo%3d
%5bvti%5ftimelastmodified%3bSW%7c08+June+2006+21%3a40%3a07+%2d0000%5d
%5d
```

Example (decoded for readability):

```
&document=
[document_name=folder1/folder2/small.txt;
meta_info=[vti_modifiedby;SW|user_name;
vti_author;SW|user_name]]
```

2.2.2.5.6 Document-Return-Type

Used by the server to return a single document and its metadata.

```
DOCUMENT-RETURN-TYPE = OBRACKET "document_name" VALSEP URL-STRING
ARGSEP "metainfo" VALSEP METADICT CBRACKET
```

2.2.2.5.7 Document-List-Return-Type

Used by the server to return a list of documents and their metadata.

```
DOCUMENT-LIST-RETURN-TYPE = OBRACKET *(OBRACKET "document_name" VALSEP
URL-STRING ARGSEP "meta_info" VALSEP METADICT CBRACKET) CBRACKET
```

2.2.2.5.8 Service-Return-Type

Used to return information about a Web site.

```
SERVER-RELATIVE-URL-STRING = URL-STRING
```

The URL must be service-relative.

```
SERVICE-RETURN-TYPE = OBRACKET "service_name" VALSEP
```

```
SERVER-RELATIVE-URL-STRING ARGSEP "meta_info" VALSEP METADICT CBRACKET
```

2.2.2.5.9 DOC-INFO Request

Used to return information about a document name and its metadata.

```
DOC-INFO-REQUEST = ARG-NAME DOCINFO
```

The RPC-KEY-STRING in ARG-NAME MUST be "document".

2.2.2.5.10 Url-Directory

Provides the name and metadata associated with a given URL.

```
URL-DIRECTORY = OBRACKET "url" VALSEP URL-STRING ARGSEP "meta_info"
VALSEP METADICT CBRACKET
```

2.2.2.5.11 Status

Used to send back a status error code.

```
STATUS-CODE = UNSIGNED-INT
STATUS = OBRACKET "status" VALSEP STATUS-CODE ARGSEP "osstatus" VALSEP
STATUS-CODE ARGSEP "msg" VALSEP STRING ARGSEP "osmsg" VALSEP STRING
CBRACKET
```

2.2.2.5.11.1 Error Codes

Error Code	Message
0x0002000C	Write error on file 'value'.
0x00020019	Cannot rename 'value' to 'value': destination already exists.
0x00090002	A file with the name 'value' already exsits. It was last modified by 'value' on 'value'.
0x00090005	The URL 'value' is invalid. It may refer to a nonexistent file or folder, or refer to a valid file or folder that is not in the current Web.
0x00090006	There is no file with URL 'value' in this Web.
0x00090007	The folder that would hold URL 'value' does not exsit on the server.
0x0009000D	A folder with the name 'value' already exisits.
0x0009000E	The file 'value' is checked out or locked for editing by 'value'.
0x0009000F	The file 'value' is not checked out.
0x0009001E	Some files have been automatically checked out from the source control repository.

Error Code	Message
0x00090023	The folder 'value' does not exist. Please create the folder and then retry the operation.

2.2.2.5.12 Put-Option

Used to define the behavior of file upload operations.

```
PUT-OPTION-VAL =/ "atomic"
```

If this flag is specified, the server does all the needed checking to ensure that all the files can be updated before changing the first one. The server MAY ignore this.<2>

```
PUT-OPTION-VAL =/ "checkin"
```

The document is checked in after it is saved. This flag is only used to support **long-term checkout** operations. Clients conforming to the FrontPage Server Extensions Remote Protocol MUST NOT send this. Servers MAY ignore this parameter if they choose not to support long-term checkout.

```
PUT-OPTION-VAL =/ "checkout"
```

Valid only if checkin is specified. Notifies the source control of the new content (checkin), but keeps the document checked out. (This is the equivalent to checking the document in, and then checking it out again.) Clients conforming to the FrontPage Server Extensions Remote Protocol defined in this document MUST NOT send this option. Servers MAY ignore this parameter.

```
PUT-OPTION-VAL =/ "createdir"
```

The parent directory is created if it does not exist. Usually the server MUST require that the parent directory of a file or folder exist; if the client sends this option, it MUST try to create the immediate parent of the file being created. For example, if folder1/folder2/file.txt is being created, the server would try to create folder1/folder2, but not folder1 if it did not already exist.

```
PUT-OPTION-VAL =/ "edit"
```

Uses the date and time the document was last modified to determine whether the item has been concurrently modified by another user. This flag is used to prevent race conditions where two users could edit the same data. If this flag is specified and the inbound modification time does not match the value on the server, the server MUST reject the upload. The client SHOULD send this flag unless a higher level has indicated it needs to overwrite changes.

```
PUT-OPTION-VAL =/ "forceversions"
```

Not used by the FrontPage Server Extensions Remote Protocol. Acts as though versioning is enabled, even if it is not. Clients conforming to the FrontPage Server Extensions Remote Protocol MUST NOT send this option. Servers MAY ignore this parameter.<3>

```
PUT-OPTION-VAL =/ "listthickets"
```

Requests that metadata be returned for **thicket** supporting files. The server MUST act as though this parameter was sent if the effective protocol version is less than 5.0.

```
PUT-OPTION-VAL =/ "migrationsemantics"
```

Not used by the FrontPage Server Extensions Remote Protocol. Preserves information about who created the file and when. Clients conforming to the FrontPage Server Extensions Remote Protocol MUST NOT send this option. The server MAY ignore this option. If the server wants to honor this option, it SHOULD do additional authorization and ignore the option if the authorization fails.<4>

```
PUT-OPTION-VAL =/ "noadd"
```

Does not add the document to source control. Clients conforming to the FrontPage Server Extensions Remote Protocol MUST NOT send this option. The server SHOULD ignore this option.<5>

```
PUT-OPTION-VAL =/ "overwrite"
```

Uses the date and time the document was last modified, as specified in the inbound metainfo, rather than the extent of time on the server.

```
PUT-OPTION-VAL =/ "thicket"
```

Specifies that the associated file is a thicket supporting file. The server SHOULD detect that the upload includes a thicket supporting file and infer this flag.

```
PUT-OPTION = *(PUT-OPTION-VAL ",") PUT-OPTION-VAL
```

The PUT-OPTION data type MUST contain at least one PUT-OPTION-VAL.

2.2.2.5.13 Rename-Option

Used to define the behaviors of a rename operation.

```
RENAME-OPTION-VAL = "createdir"
```

Creates the parent directory if it does not already exist. This flag is analogous to the "createdir" PUT-OPTION-VAL (as specified in Put-Option) and has the same semantics.

```
RENAME-OPTION-VAL = "findbacklinks"
```

Requests that servers, implementing **link fixup**, fix the linked files other than those moved. The server MAY ignore this flag.

```
RENAME-OPTION-VAL =/ "nochangeall"
```

Do not perform link fixup on links in moved documents. This parameter is used in publishing scenarios. Clients conforming to the FrontPage Server Extensions Remote Protocol MUST NOT send this option. The server MAY ignore this.<6>

```
RENAME-OPTION-VAL =/ "patchprefix"
```

Simulates the move of a directory rather than a file. Clients conforming to the FrontPage Server Extensions:Website Management MUST NOT send this option; the server SHOULD ignore this flag for the usage defined in this document.

```
RENAME-OPTION =/ "none"
/ RENAME-OPTION-VAL *("," RENAME-OPTION-VAL)
```

Note The client MUST send "none" if it does not want to specify any of the options given by a RENAME-OPTION-VAL.

2.2.2.6 Irrecoverable Error Responses

If the server encounters an error and cannot proceed with the request (for example, when a request is not syntactically valid, when the server is required to allocate more memory than it can, or when the request finds itself in some other irrecoverable failure situation), it MUST send in HTML Mode the un-escaped string:

```
"*-*-* :-| :^| :-/  :-( 8-( *-*-*"
```

followed by a STATUS. The client SHOULD recognize this sequence as an indication that the server failed in some catastrophic way and SHOULD abandon any parsing context it was in. The server MUST NOT use this alert sequence in a context where the string might otherwise be valid, such as when it is returning file contents. An example of a syntax error would be a string other than "true" or "false" given when a Boolean value is expected.

2.2.3 Entry Points

Each method call is an HTTP POST by the client to a URL on the server. There are four entry points on any server, which can be discovered by the clients (see section 4.1.1). Each method entry denotes which entry point MUST be used to call that method. The four entry points are detailed in the table below.

Name	Description
FPShtmlScriptUrl	Used to retrieve the server version method, as specified in section 3.1.5.3.14; also for the url to web url method, as specified in 3.1.5.3.16.
FPAuthorScriptURL	Used for all methods to deal with document manipulation.
FPAdminScriptURL	Not used in the FrontPage Server Extensions Remote Protocol.

Name	Description
TPScriptURL	Used by SharePoint Team Services dialogview 3.1.5.3.17.1.

2.2.4 Metainformation

Files, folders, and Web sites in servers have an associated metadictionary, which contains strings (called keys or metakeys) which are mapped to strongly typed values. These metakey-value pairs are called metadata. Server implementations use the metadict to store details about entities for later use by the server. Clients store values in metadicts for later use by the same client or other clients. A limited number of well-known metakeys are used for client/server communication. These shared metakeys are specified in this document.

The metabase (consisting of a series of metakeys, or FrontPage Server Extension settings and properties) can be found in the _VTI_CNF folder beneath each folder on the Web server in the default file.

The metabase uses the colon character to separate the metakey name from the datatype. Next, the pipe ("|") character is used to separate the datatype from the metakey value. A metakey should consist of a single line within the default file.

2.2.4.1 Type

Each metakey listed has an associated value Type, which is one of the METADICT-VALUE types defined in section 2.2.2.5.3. These are generic types which MAY be further specified in the individual metakey's description.

2.2.4.2 Client Access

The Client Access: heading refers to whether the client is able to set this **metadata** on the server.

- Read-only: Some communication from server to client is based on configuration information and Web site settings or document information which is parsed and returned to the client; the client cannot change this information. These metakeys are identified as Read-only.
- Read-write: Metadata that the client is able to set on the server is identified as Read-write.

2.2.4.3 Applies To

Metadata is associated with various entities on the server, which are identified in the Applies To: heading.

- Service: The Service value refers to metadata associated with the Server or a particular Web site.
- Folder: **Folder** refers to metadata associated with a folder, directory, or list.
- File: File refers to metadata associated with a file or document.

2.2.4.4 vti_casesensitiveurls

Type: INT

Client Access: Read-only

Applies to: Service

Description:

Contains an INT flag indicating whether the server is case insensitive with respect to URLs. If the value is 0, the default, the server is case insensitive with respect to URLs. If the value is 1, the server is case sensitive with respect to URLs.

The server SHOULD include this key as an INT in the metadata returned by the open service method.

The server MUST return 0 or 1 for this metakey.

It MUST only be 0 if URLs that differ only by case are considered equivalent.

The client SHOULD assume that the value is 0 if this key is not present.

```
Examples:

vti_casesensitiveurls:IX|0
vti_casesensitiveurls:IX|1
```

Property Value

Value	Meaning
Case-Insensitive 0	Specifies that the server is case insensitive with respect to URLs.
Case-Sensitive 1	Specifies that the server is case sensitive with respect to URLs.

2.2.4.5 vti_dirlateststamp

Type: TIME

Client Access: Read-only

Applies to: Folder

Description:

A timestamp that records the approximate time of the most recent call to the list documents method which included this folder.

The server SHOULD include this key as a date in folder metadata it returns to the client.

If the client caches the response of the list documents method requests, it SHOULD cache this timestamp, and send this value in the folderList parameter in subsequent calls to the list documents method. Servers SHOULD use the value during processing of a call to the list documents method for optimization, to only return data for folders which are out of date on the client.

If the client caches the response of the list documents method requests, it SHOULD cache this information also.

This value SHOULD be used in the *folderList* parameter when the list documents method is called again to refresh the cached response.

Example:

```
vti_dirlateststamp:TX|08+Jan+2000+19:09:27+-0000
```

Property Value

Indicates the time, which includes the date, at which the current object last had changes made to it.

Value	Meaning
vti_dirlateststamp -12 — 14	The date/time value at the end of the metakey line is the number of hours behind (to the West) or ahead (to the East) of Coordinated Universal Time (UTC). If the value is positive, it is the number of hours the local time is ahead of UTC. If the value is negative, it is the number of hours the local time is behind UTC. In the example, the number -0700 tells you the local time is 7 hours behind (or to the West) of UTC. **Note** Greenwich Mean Time (GMT) is equivalent to UTC when fractions of a second are not considered important.

2.2.4.6 vti_filesize

Type: INT

Client Access: Read-only

Applies to: File

Description:

The size of the document in bytes.

The server MUST determine the size of the file in bytes and return this value on request by the client.

The server MUST return this key as an integer in the file metadata it returns to the client. It MUST be the size of the file in bytes.

Example:

```
vti_filesize:IX|1120
```

Property Value

Indicates the current size of the current file, in bytes.

Value	Meaning
vti_filesize Filesize	The size of the currently opened file, in bytes.

2.2.4.7 vti_hassubdirs

Type: BOOLEAN

Client Access: Read-only

Applies to: Folder

Description:

The folder has subdirectories.

The server SHOULD return this key and set it to true only if the folder has subdirectories. The client MAY use this key to decide whether to display user interface elements to expand a node in a rendered directory hierarchy.

The server MUST use and maintain its own value rather than use a client-supplied value for this key.

```
Examples:

vti_hassubdirs:BR|false
vti_hassubdirs:BR|true
```

Property Value

Indicates whether or not the current directory has child folders.

Value	Meaning
False false	The current directory does not have child folders.
True true	The current directory has child folders.

2.2.4.8 vti_isbrowsable

Type: BOOLEAN

Applies to: Folder

The server SHOULD return this key and set its value to true only if the folder contents are accessible through normal HTTP requests.

The server MUST use and maintain its own value rather than use a client-supplied value for this key.

```
Example:

vti_isbrowsable:BR|false
vti_isbrowsable:BR|true
```

Property Value

Indicates whether or not the folder permits viewing the file directory of a site folder.

Value	Meaning
False	Specifies that a folder does not permit viewing the file directory.

Value	Meaning
false	
True true	Specifies that a folder allows users to view the file directory.

2.2.4.9 vti_ischildweb

Type: BOOLEAN

Client Access: Read-only

Applies to: Folder

Description:

The folder is the root of another Web site (a subweb) within this Web site.

The server SHOULD include this key on folder metadata it enumerates when the folder is the root of another service. The client MAY use this information to avoid further calls to the url to web url method when traversing a folder hierarchy that might span services. The client also MAY use this key to indicate to the user that the folder represents a service boundary.

The server MUST use and maintain its own value rather than use a client-supplied value for this key.

```
Examples:

vti_ischildweb:BR|false
vti_ischildweb:BR|true
```

Property Value

Indicates whether or not the folder is also the root directory of a subsite.

Value	Meaning
False false	Specifies that a folder is not the root directory of a subsite.
True true	Specifies that a folder is also the root directory of a subsite.

2.2.4.10 vti_isexecutable

Type: BOOLEAN

Applies to: Folder

The server SHOULD include this BOOLEAN key on folder metadata. A value of true indicates that the file or the contents of the folder may be executed, regardless of file type.

The server MUST use and maintain its own value rather than use a client-supplied value for this key.

```
Examples:
```

```
vti_isexecutable:BR|false
vti_isexecutable:BR|true
```

Property Value

Specifies whether or not the server can execute programs in the current directory.

Value	Meaning
False false	The server does not permit the execution of programs in the current directory.
True true	The server allows the execution of programs in the current directory.

2.2.4.11 vti_isscriptable

Type: BOOLEAN

Applies to: Folder

The server SHOULD include this BOOLEAN key on folder metadata. A value of true indicates that the file or the contents of the folder may be executed if they are script files or static content. A value of false only allows static files to be served.

The server MUST use and maintain its own value rather than use a client-supplied value for this key.

```
Examples:

vti_isscriptable:BR|false
vti_isscriptable:BR|true
```

Property Value

Determines whether or not the server can execute scripts in the current directory.

Value	Meaning
False false	The server does not allow the execution of scripts in the current directory.
True true	The server permits the execution of scripts in the current directory.

2.2.4.12 vti_longfilenames

Type: INT

Client Access: Read-only

Applies to: Service

Description:

A flag indicating whether the server supports long file names of up to 255 characters. If the value of this metakey is 1, the server supports long file names. If the value of this metakey is 0, the server does not support long file names.

The server MUST set this value to 0 if it only supports **short file names**; otherwise, it SHOULD set this value to 1.

The client MUST send only short file names when dealing with a server reporting 0 for this value.

The server MUST use and maintain its own value rather than use a client-supplied value for this key. <7>

```
Examples:

vti_longfilenames:IX|0
vti_longfilenames:IX|1
```

Property Value

Indicates whether or not the underlying file system of the Web server supports long file names.

Value	Meaning
False 0	Indicates that the underlying file system of the Web server does not support long file names.
True 1	Indicates that the underlying file system of the Web server supports long file names.

2.2.4.13 vti_metatags

Type: METADICT-STRING-VECTOR

Applies to: File

Description:

A list of the META element tag settings for the current document, if any.

For HTML files, the server SHOULD maintain a list of META tags in the file. If the server maintains this list, this key MUST be present and MUST have a pair of entries for each META element tag. For META element tags that have the HTTP-EQUIV attribute, the first string in the pair MUST be "HTTP-EQUIV=" followed by the value of the HTTP-EQUIV attribute; the second string in the pair MUST be the value of the CONTENT attribute. For META element tags that have NAME and CONTENT attributes, the first string in the pair MUST be the value of the NAME attribute and the second string MUST be the value of the CONTENT attribute.

A client MAY alter the way it displays files based on this value.

The server MUST parse the document for this value and MAY cache the value for return to the client on request. The client cannot set this value directly, but MAY change it by updating the document.

The server MUST use and maintain its own value rather than use a client-supplied value for this key.<8>

```
Examples:

vti_metatags:VR|HTTP-EQUIV=Content-Type text/html;\ charset=utf-16
vti_metatags:VR|HTTP-EQUIV =Content-Language en-us
```

Property Value

Indicates the metadata settings for the current web.

Value	Meaning
vti_metatags Stringvalue	Specifies the current metadata page settings for the current web.

2.2.4.14 vti_showhiddenpages

Type: INT

Applies to: Service

Description:

The server MUST have a value of 0 or 1 for this key. If the value is nonzero, then the server is indicating that it is not configured to support **hidden documents** and the client MUST send a value of true for the listHiddenDocs parameter to the list documents method; if the value is zero, the client should respect higher-layer decisions when requesting hidden documents to determine whether they should be retrieved or not.

```
Examples:

vti_showhiddenpages:IW|0
vti_showhiddenpages:IW|1
```

Property Value

Value	Meaning
False 0	Specifies that hidden pages will not be displayed in file listings.
True 1	Specifies that hidden pages will be displayed in file listings.

2.2.4.15 vti_sourcecontrolcheckedoutby

Property Value

Type: STRING

Client Access: Read-only

Applies to: File

Description:

The login name of the user who opened the page under source control.

The server MUST include this value only if a file is checked out either short-term or long-term. The server MUST record the authenticated login username of the client when a document is checked in and store it in this metakey for return to the client on request. When it is set, it MUST be a string suitable for display that identifies a user.

The value stored in this metakey MUST be comparable to vti_username. The values SHOULD be the same only if the user making the request has the file checked out.

The server MUST use and maintain its own value rather than use a client-supplied value for this key.

```
Example:

vti_sourcecontrolcheckedoutby:SX|CORPDOMAIN\\johnsmith
```

Value	Meaning
UserName	The value is the user name of the user that opened the page under source control.<9>

2.2.4.16 vti_sourcecontroltimecheckedout

Type: TIME

Client Access: Read-only

Applies to: File

Description:

The time the document was checked out on the server.

A server MUST include this value only if a file is checked out either short-term or long-term. When it is set, it MUST provide a timestamp indicating when the file was checked out.

The client cannot set this value directly, but sets it as a side-effect of a checkout method or get document method with a checkout parameter set.

The server MUST use and maintain its own value rather than use a client-supplied value for this key.

```
Example:

vti_sourcecontroltimecheckedout:TX|15 Apr 2000 12:52:03 -0000
```

Property Value

The time, including date, at which the current object was placed in a checked-out state with source control.

Value	Meaning
vti_sourcecontroltimecheckedout	The date and time that the user placed a file contained within the web

Value	Meaning
DateTime	into a checked-out state for local modification.

2.2.4.17 vti_thicketdir

Type: BOOLEAN

Applies to: File,Folder

Description:

This metakey contains different content depending on whether it appears in the metadata for a document or a folder.

Folder

When applied to a folder, this metakey contains a BOOLEAN flag specifying whether the folder contains the supporting files for a thicket.

The server SHOULD include this metakey and set its value to true for a folder that contains files supporting an HTML thicket. For folders that do not contain supporting files, the server SHOULD omit this but MAY send the key as false. The client SHOULD adapt its rendering so that it does not show folders for which this key is true.

Document

When applied to a document, this metakey contains a STRING with the name of the folder being used for supporting thicket files.

The server SHOULD include this metakey for a document which is the main file of an HTML thicket.

The server SHOULD update document and folder metakeys to indicate the presence and relationship of a HTML thicket document and its thicket supporting folder when handling the put document method with a put_option value of "thicket".

```
Example:

vti_thicketdir:SW|jobs/index_files
```

Property Value

Indicates the name of the folder being used for cache files for the WebFolder service.

Value	Meaning
vti_thicketdir FolderName	Specifies the relative name of the directory within the current web being used for cache files for the WebFolder service.

2.2.4.18 vti_thicketsupportingfile

Type: BOOLEAN

Client Access: Read-only

Applies to: File

Description:

Indicates whether or not the document is a supporting file in an HTML thicket.

The server SHOULD include this key and set its value to true for a file that is a supporting file in an HTML thicket. For files that are not supporting files, the server SHOULD omit this but MAY send the key as false. The client SHOULD adapt its rendering so that it does not show files for which this key is true.

The server SHOULD use and maintain its own value rather than use a client-supplied value for this key. It MAY consider values passed by the client or the put document PUT-OPT-VAL of "thicket".

```
Examples:

vti_thicketsupportingfile:BW|false
vti_thicketsupportingfile:BW|true
```

Property Value

Indicates whether or not the WebFolder service is in use with a cache file.

Value	Meaning
False false	Specifies that a WebFolder service cache file is currently not in use.
True true	Specifies that a WebFolder service cache file is currently being used.

2.2.4.19 vti_timecreated

Type: TIME

Applies to: File, Folder

Description:

The time the document or folder was created.

The server SHOULD include this key for files and folders. It SHOULD reflect the date and time the file or folder was created. The client MAY use this value to render details about files or folders.

The server MUST use and maintain its own value rather than use a client-supplied value for this key.

```
Example:

vti_timecreated:TR|24 Apr 2000 15:54:33 -0000
```

Property Value

The time, including date, at which the current object was created.

Value	Meaning
vti_timecreated DateTime	The creation time of the current object.

2.2.4.20 vti_timelastmodified

Type: TIME

Client Access: Read-only

Applies to: File, Folder

Description:

The time the document or folder was most recently modified.

The server SHOULD include this key for files and folders. It SHOULD approximate the date and time the file or folder was last modified. As specified in Put-Option, section 2.2.2.5.12, under the edit and overwrite values, the server MUST use this for concurrency control and thus might not be able to make this reflect the time that the file was last modified. The client MAY use this value to render details about files or folders; however, vti_timelastwritten will often be more appropriate when it is available. The client SHOULD send this value in accordance with its use, as specified in section 2.2.2.5.12, under the edit and overwrite values. <10>

```
Example:

vti_timelastmodified:TR|24 Apr 2000 15:54:33 -0000
```

Property Value

The time, including the date, at which the current object last had changes made to it.

Value	Meaning
vti_timelastmodified DateTime	For files and folders, the date and time that the object last had changes made to it.

2.2.4.21 vti_timelastwritten

Type: TIME

Client Access: Read-only

Applies to: File, Folder

Description:

The time the document or folder was last saved to local storage on the server.

The server SHOULD include this metakey for files and folders. The server SHOULD record the date and time the file or folder content was modified in this metakey. The client SHOULD use this value to render details about files or folders but MAY use vti_timelastmodified for that purpose.

The server SHOULD use and maintain its own value rather than use a client-supplied value for this key.<11>

```
Example:

vti_timelastwritten:TR|24 Apr 2000 15:54:33 -0000
```

Property Value

The time, including the date, at which the current object last was saved to disk.

Value	Meaning
vti_timelastwritten DateTime	For files and folders, the date and time of the last file modification.

2.2.4.22 vti_title

Type: STRING

Client Access: Read-write

Applies to: File, Service

Description:

The user-readable title of a document or Web site.

The server SHOULD maintain this key for a Web site as a user-readable description of the Web site. The client MAY use this string to refer to the Web site when presenting information to a user.

The server SHOULD maintain this key for documents. If the document is an HTML document on the server, the server SHOULD parse the document for the content of a TITLE element tag, and MAY cache this value for return to the client. The client MAY update this metakey for HTML documents and the server SHOULD rewrite the document with an updated TITLE element.

For file streams which the server is unable to parse, the server SHOULD accept a client-supplied metakey value and return it on client request.

The client SHOULD use this metakey value where the title of the document is to be displayed to the user.

```
Example:

vti_title:SR|Trey Research
```

Property Value

Can be either a string that is the title of the web page or, in some cases, identifies the location of the title on the page.

Value	Meaning
vti_title PageTitle	The title of the web page.

2.2.4.23 vti_username

Type: String

Client Access: Read-only

Applies to: Service

Description:

The current authenticated login username associated with the client.

The server SHOULD include this metakey in the Web site metadata. When the key is present, the client SHOULD use this key for comparisons with vti_sourcecontrolcheckedoutby.

The server SHOULD include this key in the service metadata. Its value should be a string that uniquely identifies the user to the server. When the key is present, the client SHOULD use this key rather than any information it passed to the HTTP layer for authentication for comparison with vti_sourcecontrolcheckedoutby.

The server SHOULD use and maintain its own value rather than use a client-supplied value for this key. <12>

```
Example:

vti_username:SX|CORPDOMAIN\\johnsmith
```

Property Value

Identifies the logged-in user.

3 Protocol Details

The following sections specify abstract data model, entry points, and message processing rules.

3.1 Server Details

A Microsoft FrontPage Server Extensions client SHOULD initialize a connection with the server. The client MAY then make as many method calls against the server as needed. The FrontPage Server Extensions Remote Protocol, like HTTP 1.1, as specified in [RFC2616], is a stateless protocol. As such, connections do not need to be closed.

The SharePoint Team Services dialogview client of the FrontPage Server Extensions Remote Protocol is simply used for viewing information on a server. After the dialogview connection is initialized, the client MAY make any supported calls against the server. The FrontPage Server Extensions Remote Protocol connections do not need to be closed.

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

3.1.1 Abstract Data Model

The FrontPage Server Extensions Remote Protocol is used to communicate with remote file systems for administrative purposes. For those reasons, a data model that looks like a file system is useful.

The FrontPage Server Extensions Remote Protocol has three concepts in its file system: files, directories, and services. Each of these structures MUST have a dictionary of metadata associated with it. This metadata can be used by clients or servers to store whatever information is relevant to the object. The metadata dictionary is also often used as a way for the server to communicate information about a file to the client, such as its checkout state.

In addition to the metadata dictionary, here are the relevant properties of each file system concept:

1. Files MUST have a content stream, that is, an array of sequenced bytes, that represent the contents of the file.

2. Directories MUST not have a content stream, but MUST be able to contain files or other directories.

3. Services model the concept of a Web site. Like directories, they MUST not have a content stream and MUST be able to contain files or directories. Services MUST also be able to answer questions about their own capabilities, for example, if they support source control. The capabilities of a service are communicated to clients by using the metadata dictionary of the service.

3.1.1.1 Source Control

Servers MUST support **short-term checkout** and SHOULD implement a source control sandbox. The server MAY offer users the option to turn the source control sandbox off.<13>

For a server that has the source control sandbox turned off: When a document is checked out, the server MUST refuse to perform any operations on the document that are not sent by the user who has the document checked out, including another user requesting to read the document.

For a server that has the source control sandbox turned on: When a document is checked out, the server MUST refuse any requests to modify the document that are not sent by the user who has the

document checked out. However, if another user merely requests the contents of the document, the server SHOULD respond with the document stream as it appeared when the document was checked out.

A document that is checked out short-term can have the checkout released in either of two ways: The client can send an uncheckout document request to the server or the short-term checkout expires.

3.1.2 Timers

3.1.2.1 Short-Term Checkout Timer

A document checked out short-term has a time-out value associated with the checkout. The length of the short-term checkout timer is determined based on client input, as detailed in the checkout document method, section 3.1.5.3.2.<14>

3.1.3 Initialization

Microsoft FrontPage Server Extensions initialization requires the following operations to take place.

3.1.3.1 Determining Server Capabilities

A prospective client MUST perform an HTTP OPTIONS request, as specified in [RFC2616] section 9.2, against a server to determine if it is a FrontPage Server Extensions Remote Protocol server.

When receiving an OPTIONS request, a FrontPage Server Extensions Remote Protocol server MUST return the header 'MS-Author-Via:' with a value that includes 'MS-FP/4.0' to indicate that the server supports the FrontPage Server Extensions Remote Protocol. The client SHOULD try the protocols listed in the MS-Author-Via header in the order they appear in the header.

When receiving an OPTIONS request, a server that supports SharePoint Team Services dialogview MUST return the header '50_Collab' to indicate that the server supports the dialogview. Results of the HTTP OPTIONS request that are returned from the server SHOULD be cached by clients as a performance optimization.

3.1.3.2 Determining Entry Points

3.1.3.2.1 Client Request for Entry Point HTML Page

If the server supports the FrontPage Server Extensions Remote Protocol, the client SHOULD perform an HTTP GET of vti_inf.html at the Web (server) root by using the following to determine the entry point for the Microsoft FrontPage Server Extensions: <15>

```
http://fpseserver/_vti_inf.html
```

where fspeserver is the Web root of the server.

3.1.3.2.2 Server Entry Point HTML Page Response

The server MUST reply to the HTTP GET of vti_inf.html with an HTML page containing an HTML comment that is an ENTRY-POINT-COMMENT, as defined below:

```
ENTRY-POINT-COMMENT = COMMENT-BEGIN + SHTML-ENTRY-POINT +
```

```
AUTHOR-ENTRY-POINT + ADMIN-ENTRY-POINT + TPSCRIPT-ENTRY-POINT +
COMMENT-CLOSE

COMMENT-BEGIN = "<!-- FrontPage Configuration Information FPVersion="
+ DQUOTE + VERSION + DQUOTE + LF

SHTML-ENTRY-POINT = "FPShtmlScriptUrl=" + DQUOTE +
SERVICE-RELATIVE-URL + DQUOTE + LF

AUTHOR-ENTRY-POINT = "FPAuthorScriptUrl=" + DQUOTE +
SERVICE-RELATIVE-URL + DQUOTE + LF

ADMIN-ENTRY-POINT = "FPAdminScriptUrl=" + DQUOTE +
SERVICE-RELATIVE-URL + DQUOTE + LF

TPSCRIPT-ENTRY-POINT = "TPScriptUrl=" + DQUOTE +
SERVICE-RELATIVE-URL + DQUOTE + LF
```

Servers SHOULD return a comment that defines the entry points as below, as clients MAY assume these values: <16>

```
<!-- FrontPage Configuration Information FPVersion="12.0.0.000"
FPShtmlScriptUrl="_vti_bin/shtml.dll/_vti_rpc"
FPAuthorScriptUrl="_vti_bin/_vti_aut/author.dll"
FPAdminScriptUrl="_vti_bin/_vti_adm/admin.dll"
TPScriptUrl="_vti_bin/owssvr.dll" -->
```

For descriptions of the fields used in the HTML comment, see section 2.2.3. Each method description contains a section that defines which entry point that method must use. Clients MUST post to the correct entry point, or the server SHOULD ignore their request.

The client SHOULD then call the server version method on the root of the server to determine the latest (highest) server version of the protocol that the server supports. The client MUST choose the earlier (lower) version supported between its latest (highest) protocol version and the server's latest (highest) protocol version for use during the connection.

If the client is opening a Web site that is not at the root of the server, the client MUST call the url to web url request. This method accepts a **server-relative** URL, such as "/subweb/folder/document.txt". It would return the server-relative URL of the Web site, "/subweb", and the service-relative URL of the item, "folder/document.txt".

The client MUST then post all further methods to the Web site that it wants to communicate with. The *service_name* parameter MUST NOT be used by the client to denote what Web site it is communicating with, as this parameter is ignored by the server. For example, if the client wants to check out "/subweb/folder/document.txt", it needs to post to the SHTML-ENTRY-POINT of the subweb. Assuming the default value of the SHTML-ENTRY-POINT, that would be:

```
http://fpseserver/subweb/_vti_bin/shtml.dll/_vti_rpc
```

The client then refers to the file it wants to check out by its service-relative URL within the FrontPage Server Extensions Remote Protocol request. Finally, the client SHOULD call the open service method on the appropriate Web site to begin the conversation. The client MAY then make whatever method calls it needs against the server.

The SharePoint Team Services dialogview client SHOULD perform an HTTP HEAD request,as specified in [RFC2616], section 9.4 against owssvr.dll (for details, see section 3.1.3). If the return code from the server is 200, the client SHOULD perform an HTTP GET, as specified in [RFC2616], section 9.3 to retrieve the full page body. If the return code from the server is 410, the client MUST NOT perform an HTTP GET on the server.

3.1.4 Higher-Layer Triggered Events

There are no higher-layer triggered events for the FrontPage Server Extensions Remote Protocol server. Each client request is triggered by the client application's needs.

3.1.5 Message Processing Events and Sequencing Rules

Aside from initialization, methods MAY be called in any order, as determined by the client application's needs. The only methods forming a strong logical pair are the checkout document and uncheckout document methods. Clients that call the former SHOULD call the latter because a document that is checked out cannot be edited by any other users until the checkout is revoked.

Protocol Examples, section 4, provides examples that show common operations being performed against the server, giving some guidance about common method sequences.

3.1.5.1 HTTP Headers

The client SHOULD send an X-Vermeer-Content-Type header, as specified in [RFC2616], section 14.17 with the same value as the standard HTTP Content-Type header to safeguard against one-click attacks (see section 5.1). The server MUST use this header, if present, to determine the Content-Type of the request. If this header is not present, the server SHOULD fail the request.<17>

Clients MUST also include the string "FrontPage" (case sensitive) in its User-Agent header, as specified in [RFC2616], section 14.43. The server MAY alter its responses when the client does not do this.<18>

Except as specified in specific methods, server responses MUST have the HTTP Content-Type "application/x-vermeer-rpc".

3.1.5.2 Method Formatting

A set of formatting conventions are used for each Microsoft FrontPage Server Extensions method specified in the Methods section later in this document.

The Methods section begins with a brief description of the method's purpose. The Tokens section follows the description and corresponds to the REQUEST element that specifies the set of arguments for each method. Clients MAY pass in any subset of the given arguments, although some arguments are required. Clients MAY pass the arguments in any order, and servers MUST accept them in any order. Clients SHOULD NOT pass arguments unless they are mentioned in this document. When a server is passed an argument it does not recognize, the server SHOULD treat it as a syntax error. It MAY choose to ignore the parameter instead.<19>

Each argument definition starts with an argument name in bold type. The argument name corresponds to the ARG-NAME element. The data type is listed in the argument description, and it defines the required format for the ARG-VALUE element.

The URL section defines the URL as provided by _vti_inf.html to which clients MUST post when using this method.

The Return Values section specifies the set of return values, and its formatting is similar to the Fields section. The RET-NAME corresponds to the return value name in bold type. The format of the RET-VALUE is defined by the return value type. Servers MAY return arguments in any order, and clients MUST accept the parameters in any order.

In error conditions at the FrontPage Server Extensions Remote Protocol level, the server SHOULD return a RET-NAME RET-VALUE pair where the RET-NAME is "status" and the RET-VALUE is a STATUS object (see 2.2.2.5.11). A client SHOULD ignore all other return values if a status is present. Even though this is an error, it SHOULD be encapsulated in an HTTP 200 response, as specified in [RFC2616] section 10.4.2.

Lower layers (such as HTTP or IP) may also return errors. For example, the FrontPage Server Extensions Remote Protocol layer on the server SHOULD indicate to the HTTP layer that it requires authentication, which in turn SHOULD cause the HTTP layer to send a 401 message in response, as specified in [RFC2616] section 10.4.2, to unauthenticated requests.

If the lower layers pass on an unauthenticated request from the client to the FrontPage Server Extensions Remote Protocol server, it SHOULD respond with an HTTP 401 as specified in [RFC2616] section 10.4.2. It MAY include an entity body containing a descriptive error message in the response.<20>

3.1.5.3 Methods

Each request by a client that uses the FrontPage Server Extensions Remote Protocol MUST begin with a field that contains the method name. For example, the get document request has the string "get document" in the method field. Following the method name field is the list of arguments that MAY be specified in any order.<21>

The client POSTs the following FrontPage Server Extensions Remote Protocol requests to the server. These requests are specified in the sections that follow.

3.1.5.3.1 Common Method Arguments

The following argument values are common to many of the methods.

Tokens

validateWelcomeNames: A Boolean value that determines if each item in the file's list of links SHOULD replace links to directories with links to the welcome page (**default Web page**) for that directory, if it exists. If true, checking for default names is done; if false, checking is not done. This parameter MUST NOT be passed by a client that conforms to the FrontPage Server Extensions Remote Protocol. The server MAY ignore this value.

listLinkInfo: A Boolean value that specifies if the server response SHOULD contain information about the links from the current page or pages. If true, the response SHOULD include link information; if false, link information SHOULD be excluded to reduce bandwidth and server processing overhead. Passing any other string causes a syntax error. Clients conforming to the FrontPage Server Extensions Remote Protocol MUST send false for this parameter; servers MAY ignore this parameter and avoid sending link information for non-conforming clients.<22>

service_name: This parameter is obsolete, but the client MAY send this URL-STRING defining the server-relative URL of the Web site that the request is addressed to. However, the server MUST ignore it. The URL to which the request is posted MUST be used to determine what Web site the request is issued against.

return_stats: The client MUST NOT send this Boolean value. If the server receives it, the server MUST validate its data type but SHOULD otherwise ignore it. If a server chooses not to ignore it, it MAY return an additional return value that gives details about the cost of the request.<23>

3.1.5.3.2 checkout document

The checkout document request is used by the client to enable the currently authenticated user to make changes to a document under source control.

Tokens

service_name: This parameter is deprecated; see service_name in section 3.1.5.3.1.

document_name: The client MUST send and the server MUST interpret this URL-STRING as the service-relative path of the document to check out.

force: This parameter is an INT bitmask; bits 0 and 1 are defined below. The client MUST set all unused bits to zero; the server MUST ignore unused bits.

Mask	Meaning
Bit 0 0x00000001	Force checkout. This feature is currently unimplemented and reserved for future use. Clients MUST NOT set this bit, and servers MUST ignore it.
Bit 1 0x00000002	Refresh short-term checkout. The client MUST set this bit if, and only if, it already has a short-term checkout on the file and wants to extend the time-out on it. The server MUST attempt to create a new short-term checkout if this bit is cleared, and it MUST attempt to extend an existing short-term checkout if this bit is set. It MUST return an error if this bit is set when no short-term checkout exists; it also MUST return an error if this bit is not set and a short-term checkout does not exist.

The client MUST set all unused bits to zero, and the server MUST ignore all unused bits. Range: from -2147483648 through 2147483647; only 0 and 2 are currently allowed.

timeout: An INT that defines the number of minutes that a short-term checkout is requested. To retain the lock, the client MUST renew its short-term checkout within this interval. The client MUST NOT send negative values, and the server MUST ignore them. The value zero is reserved for server implementations that understand long-term checkout and thus MUST NOT be sent by clients. Servers MAY ignore requests in which this parameter is set to zero. Range: 1 to 2147483647.

validateWelcomeNames: For semantics, see section 3.1.5.3.1.

URL

FPAuthorScriptUrl

Return Values

meta_info

A METADICT containing information about the document that has been checked out.

3.1.5.3.3 create url-directories

The create url-directories request allows the client to create one or more directories (folders) on the Web site. This operation is not atomic. If the bulk operation fails, some of the earlier directories might have been created. In the case of a failure, the client SHOULD query the server with list documents or if it needs to determine what folders were created.**Note** Clients SHOULD use this method rather than create url-directory.

Tokens

service_name: This parameter is deprecated: For semantics, see service_name in section 3.1.5.3.1.

urldirs: A VECTOR-URL-DIRECTORY specifing the names and metainformation of folders to create. The client MUST specify one URL-DIRECTORY for each folder it wants to create. The server SHOULD create the directories in the order specified. The client MUST send this parameter.

URL

FPAuthorScriptURL

Return Values

message

A STRING description of the action taken by the server. This is intended for debugging and SHOULD be ignored by the client.

3.1.5.3.4 create url-directory

The create url-directory request is used by the client to create a folder for the current Web site.**Note** Clients SHOULD use create url-directories instead of this method. However, servers MUST support this method for backward compatibility.

Tokens

service_name: This parameter is deprecated: For semantics, see service_name in section 3.1.5.3.1.

url: A URL-STRING specifying the URL of the directory to be created. The client MUST send this parameter.

executable: A Boolean indicating if the security settings for the newly created directory should allow execution of entities within it. If true, the request is to create an **executable folder**. Clients that conform to the FrontPage Server Extensions Remote Protocol MUST send false. Servers MUST ignore this for architectures that do not support the notion of an executable directory. The server SHOULD ignore this for security reasons.<24>

URL

FPAuthorScriptURL

Return Values

message

A STRING description of the action taken by the server. This is intended for debugging, and SHOULD be ignored by the client.

urldir

A URL-DIRECTORY containing the metainformation for the directory that was created.

3.1.5.3.5 getDocsMetaInfo

The getDocsMetaInfo request is used by the client to retrieve metainformation for the files and directories in the current Web site. The getDocsMetaInfo request is similar in function to the list documents request except that it only retrieves metainformation for the files or folders specified through the *url_list* parameter. The list documents requests information about all of the files and folders under a given directory; getDocsMetaInfo requests information only about the URLs requested.

Tokens

service_name: This parameter is deprecated: See service_name in section 3.1.5.3.1.

listHiddenDocs: A Boolean value that specifies if hidden documents in a Web site will be listed. If true, the server MUST list hidden documents; if false, they MUST NOT be listed.

listLinkInfo: For semantics, see section 3.1.5.3.1.

url_list: A VECTOR-STRING list of service-relative URLs about what client wants information. If url_list is empty, it is treated as though it is a single vector with "" (the root of the web site).

validateWelcomeNames: For semantics, see section 3.1.5.3.1.

URL

FPAuthorScriptURL

Return Values

document_list

A DOCUMENT-LIST-RETURN-TYPE specifying the name and metainformation for the set of documents.

urldirs

A VECTOR-URL-DIRECTORY containing information about folders and subwebs in the current Web site.

3.1.5.3.6 get document

The get document request SHOULD be used by the client to retrieve a document and its metainformation for viewing or editing on a client.

Tokens

service_name: This parameter is deprecated: For semantics, see service_name in section 3.1.5.3.1.

document_name: A URL-STRING that the client MUST send, and the server MUST interpret this value as the service-relative path of the document to retrieve. The client MUST send this parameter.

effective_protocol_version: A VERSION provides a way for the client to supersede the version sent as the PROTOCOL-VERSION-STRING in the METHOD-KEY-VALUE of the request. For details, see section 2.2.2.2. A client conforming to the FrontPage Server Extensions Remote Protocol MUST NOT send this parameter. A server SHOULD ignore this parameter.

old_theme_html: A Boolean value that determines how **theme elements** are rendered on a page. If true, the page is returned using HTML that is compatible with effective_protocol_version. If false, the effective_protocol_version corresponds to Microsoft FrontPage 97 or an earlier version of FrontPage that does not support themes. In this case, the theme information disappears, and the theme elements are rendered in the HTML used on the page. A client conforming to the FrontPage Server Extensions Remote Protocol MUST send false (either by omitting the parameter and taking the default or by explicitly sending false). A server MAY assume this parameter to be false.

expandWebPartPages: A Boolean value reserved for future use. This parameter MUST be ignored by the server regardless of a true or false value. The client MUST send false for this parameter, either explicitly or by omitting the parameter.

force: A Boolean value reserved for future use. This parameter MUST be ignored by the server, regardless of a true or false value. The client MUST send false for this parameter, either explicitly or by omitting the parameter.

doc_version: A STRING containing the source control version number of the document being retrieved. An empty string (the default value) is a request for the current version of the document. A client conforming to the FrontPage Server Extensions Remote Protocol MUST omit this parameter or explicitly send the empty string. A server MAY ignore this parameter and always send the most recent version of the document.

get_option: A STRING value that determines how documents are checked out of source control. Passing any string not listed in the following table is considered the same as "none".<25>

Value	Meaning
none	Do not check out the file.
chkoutExclusive	Check out the file exclusively. The server MUST return an error if this flag is passed and the file is already checked out by another user.
chkoutNonExclusive	Check out the file non-exclusively, if the source control system is configured to allow non-exclusive checkouts. If it is not, the server MUST treat this as chkoutExclusive instead.

The checkout MUST logically occur before the server begins sending the document to the client. If the checkout cannot occur, the server MUST send a failure and not return the document.

timeout: An UNSIGNED-INT that specifies the number of minutes the server MUST retain the short-term checkout. To retain the lock longer, the client MUST renew its short-term checkout within this interval. For details, see checkout document, section 3.1.5.3.2. The client MUST NOT send negative values, and the server MUST ignore them. The value zero is reserved for server implementations that understand long-term checkout and thus MUST not be sent by

clients. Servers MAY ignore requests in which this parameter is set to zero. Range from 1 through 4294967296.

validateWelcomeNames: For semantics, see section 3.1.5.3.1.

URL

FPAuthorScriptUrl

Return Values

message

A STRING description of the action taken by the server. This is intended for debugging, and SHOULD be ignored by the client.

document

The document stream that is sent after the </HTML> tag at the end of the standard response. This is a DOCUMENT-RETURN-TYPE, see section 2.2.2.5.6.

3.1.5.3.7 get documents

The get documents request is used by the client to retrieve a set of documents for viewing on a client computer.

Tokens

service_name: This parameter is deprecated: See service_name in section 3.1.5.3.1.

url_list: A VECTOR-STRING list of service-relative URLs specifying what documents will be retrieved by the current request.

effective_protocol_version: This parameter has the same semantics as effective_protocol_version in get document.

old_theme_html: This parameter has the same semantics as old_theme_html version in get document.

expandWebPartPages: A BOOLEAN reserved for future use. This parameter MUST be ignored by the server regardless of a true or false value. The client MUST send false for this parameter either explicitly or by omitting the parameter.

validateWelcomeNames: For semantics, see Common Method Arguments.

URL

FPAuthorScriptUrl

Return Values

This request MUST return a multipart/mixed MIME document, as specified in [RFC1341], and the responses MUST be in URL Mode rather than HTML Mode.

Each part of the response MUST be one of the following three types: ReturnValues, DocInfo, or DocData. These types are defined and scoped to this section; they exist merely to provide convenient names for the concepts.

The response MUST begin with a UrlArgs part. After the first part, each part determines the type of the next part as illustrated below.

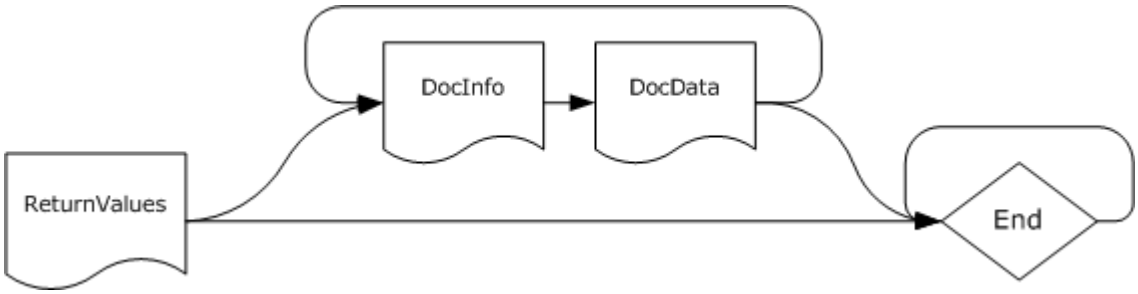

Figure 3: Type encoding sequence for POST

The UrlArgs part of the response MUST have a content type of application/x-www-form-urlencoded and MUST have a METHOD-KEY-VALUE whose REQUEST-NAME-STRING is "get documents" followed by a RET-NAME/RET-VAL pair whose RPC-KEY-STRING is "current_time" and whose RPCVALUE is a TIME.

```
"method" VALSEP "get documents" ":" PROTOCOL-VERSION-STRING
ARGSEP "current_time" TIME
```

The UrlArgs part MUST either be followed by a DocInfo part or the response MUST end.

DocInfo

The DocInfo part MUST be application/x-www-form-urlencoded and be a DOCINFO. Each DocInfo part must be followed by a DocData part.

DocData

The DocData part MUST have a Content-Type of application/octet-stream. It MUST contain the stream of the document corresponding to the previous DOCINFO part. Each DocData part MUST be followed by a DocInfo part or the response MUST terminate.

Note The mf-file-status key MUST be added to the METADICT returned in the response. If it is nonzero, the remainder of the response SHOULD be discarded by the client. This key is not considered metadata about the file, but rather metadata about the transport, which the client SHOULD examine and SHOULD remove before passing on the METADICT to higher layers.

3.1.5.3.8 list documents

The list documents request is used by the client to request a list of files, folders, and Web sites contained in a given folder.

Tokens

service_name: This parameter is deprecated: See service_name in section 3.1.5.3.1.

folderList: The keys of the METADICT are folder names and the corresponding values are the latest date for which the client has metadata. The client SHOULD send this if it has cached this information. If it is present and not empty, the values in the METADICT MUST be dates. If a folder's time stamp is missing, the server MUST return the contents of the folder with the complete metainformation. Otherwise, the server MUST return complete metainformation for

only those files, folders, and subwebs that have changed since the time stamp recorded in the folder list. For files, folders, and subwebs that have not changed, the server SHOULD return an empty METADICT to indicate that the client's cache is still valid. The client MUST interpret an empty METADICT as an indication of validity.

The following example of a value for the *folderList* parameter requests full metainformation for files in the root folder of the Web site and in the images folder that have a time stamp later than the specified dates and times. For other files in these folders, only the file name and an empty METADICT are included in the response. For details about the Dictionary and MetaDictionary, see section 2.2.2.5.3.

```
[;TX|06 Apr 2006 20:03:02 -0000;images;TX|05 Apr 2006
20:03:02 -0000]
```

initialUrl: A URL-STRING containing the URL of the folder from which documents are listed. This MUST be a service-relative URL.

listBorders: A Boolean value that specifies if the contents of the shared borders directory that contains shared border pages SHOULD be listed. If true, the contents SHOULD be listed; if false, they SHOULD NOT be listed.

listChildWebs: A Boolean value that specifies if the server response SHOULD include the names of the child Web site folders in the urldirs return value. If true, the names SHOULD be included; if false, they SHOULD NOT be included. If not set, the names of child Web site folders SHOULD NOT be included.

listDerived: A Boolean value that specifies if the list of files in the **derived documents folder** SHOULD be included in the response. If true, the list of files SHOULD be included as part of the response; if false, they SHOULD NOT be included.<26>

listExplorerDocs: A Boolean value that specifies if **task-list files** (_vti_pvt/_x_todo.htm and _vti_pvt/_x_todoh.htm) are listed. If true, the files SHOULD be listed. If false, they SHOULD NOT be listed.<27>

listFiles: A Boolean value that specifies whether the client requests information about the files in each directory that appears in the response. If true, the server MUST include the document_list return value; if false, the server MUST exclude the document_list return value.

This Boolean defaults to true; if the client does not send the parameter, the server MUST assume it to be true.

listFolders: A Boolean value that specifies if the server response SHOULD include the names and metainformation of folders under the URL specified by the *initialUrl* parameter. If true, the directory names and metainformation SHOULD be included; if false, they SHOULD NOT be included. The default value is true unless the client specifies otherwise.

This Boolean defaults to true; if the client does not send the parameter, the server MUST assume it to be true.

listHiddenDocs: A Boolean value that specifies if hidden documents in a Web site SHOULD be listed. If true, the documents SHOULD be listed; if false, they SHOULD NOT be listed.

listIncludeParent: A Boolean value that specifies if an entry for the initialUrl field SHOULD be included in the server response. If true, the entry SHOULD be included; if false, it SHOULD NOT be included.

listLinkInfo: For semantics, see section 3.1.5.3.1. This Boolean defaults to true; if the client does not send the parameter, the server MUST assume it to be true.

listRecurse: A Boolean value that specifies if the server response SHOULD recursively list the subfolders of folders under the URL specified by the *initialUrl* parameter. If true, the subfolders SHOULD be listed; if false, they SHOULD NOT be listed. The default value is true unless the client specifies otherwise. If listRecurse is set, then all children of that folder SHOULD be taken into account; otherwise, only the immediate children SHOULD be considered. This Boolean defaults to true; if the client does not send the parameter, the server MUST assume it to be true.

listThickets: A Boolean value that specifies if thicket supporting files and folders SHOULD be included in the server response. If true, the supporting files and folders SHOULD be included; if false, they SHOULD NOT be included. The default value is true unless the client specifies otherwise. This Boolean defaults to true; if the client does not send the parameter, the server MUST assume it to be true.

platform: A STRING specifying the operating system of the client that controls the listing of open **bots** (custom components). If the field is missing or empty, the server MUST NOT include the bot_list return value in the response. If the field contains data, information about any open bots on the server MAY be included. This parameter MUST NOT be passed by clients conforming to the FrontPage Server Extensions Remote Protocol.<28>

validateWelcomeNames: For semantics, see section 3.1.5.3.1.

URL

FPAuthorScriptUrl

Return Values

document_list

A DOCUMENT-LIST-RETURN-TYPE the server MUST omit this return value if the *listFiles* parameter was sent as false. If this value is returned, the server MUST list the names and metainformation for the requested set of documents specified by the parameters of the request.

bot_list

A DOCUMENT-LIST-RETURN-TYPE return value MUST NOT be included when the *platform* parameter is empty (including when the parameter is not sent). Because clients conforming to the FrontPage Server Extensions Remote Protocol MUST NOT send this parameter, the server need not support this return value. The server MAY return an empty bot_list if it wants to partially support this value.<29>

urldirs

A VECTOR-URL-DIRECTORY containing the names and metadata about folders and root directories of subwebs. The server MUST omit this return value if the *listFolders* parameter was sent as false. If this value is returned, the server MUST enumerate the folders and Web sites specified by parameters of the request.

3.1.5.3.9 move document

The move document request is used by the client to change the URL of a selected document or folder in the Web site. Note that moving a document will change its URL.

Tokens

service_name: This parameter is deprecated: See service_name in section 3.1.5.3.1.

oldUrl: A URL-STRING specifing the former URL for a document or folder whose URL is to be changed. The client MUST send this parameter. It MUST be service-relative.

newUrl: A URL-STRING specifing the new URL for a document or folder whose URL is to be changed.

url_list: A VECTOR-URL-STRING list of service-relative URLs of documents whose links should be considered for link fixup purposes. This is a hint passed from the client to the server. Servers that implement link fixup SHOULD NOT rely on the client sending the correct list. Clients that conform to the FrontPage Server Extensions Remote Protocol MUST send an empty list (either explicitly or by omitting the parameter and taking the default). Servers MAY ignore this parameter.

rename_option: This tells the server to change certain behaviors during the rename or copy operation, as defined in Rename-Option.

put_option: A set of flags describing how the operation should behave as detailed in Put-Option. In particular, the server MUST overwrite an existing file or folder if, and only if, the "overwrite" flag is added.

docopy: A BOOLEAN value that specifies if the move document request SHOULD copy or move a file to the destination. If true, the file SHOULD be copied; if false, the file SHOULD be moved. The default value is false.

validateWelcomeNames: For semantics, see section 3.1.5.3.1.

URL

FPAuthorScriptUrl

Return Values

message

A STRING description of the action taken by the server. This is intended for debugging, and SHOULD be ignored by the client.

oldUrl

A URL-STRING specifing the former URL for a document whose name or directory has changed.

newUrl

A URL-STRING specifing the new URL for a document whose name or directory has changed.

document_list

A DOCUMENT-LIST-RETURN-TYPE specifing the set of documents whose metadata has changed as a byproduct of the move due to fixing links.

moved_docs

A DOCUMENT-LIST-RETURN-TYPE of URLs and metadata for moved or copied files. Clients MAY assume and servers MUST ensure that this return value is equivalent to the document_list return

value of the list documents method, section 3.1.5.3.8, assuming the following parameter values were passed into list documents, section 3.1.5.3.8:

moved_dirs

A VECTOR-URL-DIRECTORY of URLs and metadata for moved folders. Clients MAY assume and servers MUST ensure that this return value is equivalent to the urldirs return value of the list documents method, section 3.1.5.3.8, assuming the following parameter values were passed into list documents, section 3.1.5.3.8:

initialUrl

The value that the client passed to the *newUrl* parameter of the current call to move document.

listRecurse

true

includeParent

true

As with the moved_doc return value above, the client SHOULD use this result to update any metadata cache it is maintaining.

3.1.5.3.10 open service

The open service request is used to provide Web site metainformation to the client computer.

Tokens

service_name: This parameter is deprecated: See service_name in section 3.1.5.3.1.

effective_protocol_version: This parameter has similar semantics to effective_protocol_version in get document. It is used in publishing scenarios in which the client passes the version of the intended target server. A client conforming to this subset of the protocol MUST NOT send this parameter. A server SHOULD ignore this parameter.<30>

URL

FPAuthorScriptUrl

Return Values

service

A SERVICE_RETURN_TYPE specifying the service name and Web site metainformation.

3.1.5.3.11 put document

The put document request is used by the client to write a single file to a directory in an existing Web site.

Tokens

service_name: This parameter is deprecated: See service_name in section 3.1.5.3.1.

document: A DOCINFO specifying the name and metainformation of the file to write to the directory. The client MUST send this parameter.

put_option: A set of flags describing how the operation behaves; see Put-Option for specific semantics for the options.

comment: A STRING that provides a checkin comment for the file being uploaded. The server MUST ignore this parameter unless the "checkin" PUT-OPTION-VAL is specified in the *put_option* parameter. Because clients conforming to the FrontPage Server Extensions Remote Protocol MUST NOT send the "checkin" PUT_OPTION_VAL, as specified in Put-Option, servers MAY ignore this parameter.

keep_checked_out: A BOOLEAN value that SHOULD determine a specified document's behavior in source control. If true, the document SHOULD be checked in to source control and immediately checked back out; if false, the document SHOULD be checked in. The server MUST treat this as equivalent to the "checkout" PUT-OPTION-VAL, as specified in Put-Option. Clients conforming to the FrontPage Server Extensions Remote Protocol MUST send false, either explicitly or by omitting this parameter.

validateWelcomeNames: For semantics, see section 3.1.5.3.1.

URL

FPAuthorScriptUrl

Return Values

message

A STRING description of the action taken by the server. This is intended for debugging, and SHOULD be ignored by the client.

document

A DOCINFO containing the name and metainformation of the document as it was saved. Note that even though the return value is called "document", it does not contain the document stream. The server MAY update the metainformation when the document is saved. The server MUST return the updated metainformation in the document return value.

Note The document contents MUST be sent right after the parameters in the server response. Because arguments always end with an LF in URL Mode, there is no need for further delimiters. The document stream should not be encoded in any way, as whatever is sent over the wire will be directly stored as the file contents, without further translation.

3.1.5.3.12 put documents

The put documents request is be used by the client to write multiple files to a Web site. This request SHOULD be used when a higher level wants to save a document that contains other files (such as graphics) from an application directly to a Web site directory.

This request is different from other FrontPage Server Extensions Remote Protocol requests because it receives multiple streams. The following details show how the server MUST parse the method.

The HTTP POST MUST be a multipart/mixed MIME document. Each part of the POST MUST be one of the following four types: UrlArgs, DocInfo, DocData, or End. These types are defined and scoped to this section; they exist merely to provide convenient names for the concepts.

The response MUST begin with a UrlArgs part. After the first part, each part determines the type of the next part, as illustrated below.

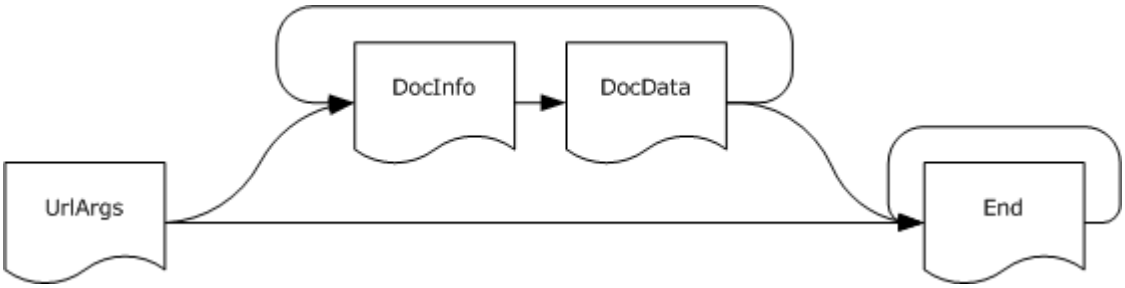

Figure 4: Type encoding sequence for POST

The UrlArgs part MUST have a content-type of application/x-www-form-urlencoded, as specified in [RFC2616] section 14.17, and MUST be parsed like a normal method request. The REQUEST-NAME-STRING, as specified in [RFC4234], MUST be put documents and it accepts the list of parameters given in the Tokens section below. The server SHOULD fail with a "client-too-old" error if the client is an earlier version than the server supports, just like any other method.

If the next part exists, it MUST be DocInfo with a content type of application/x-www-form-urlencoded, as specified in [RFC2616] section 14.17, or it MUST be End with a content type of text/html.

DocInfo: The server SHOULD parse the DocInfo part as a DOCINFO (for details, see DOC-INFO Request. The next MIME part MUST be DocData.

DocData: A DocData part MAY have any content type and MUST be the stream corresponding to the DOCINFO sent in the prior DocInfo MIME part. If the next part exists, it SHOULD be DocInfo if its content type is application/x-www-form-urlencoded, as specified in [RFC2616] section 14.17, or SHOULD be End if its content type is text/html, also specified in [RFC2616].

End: An End part MUST be ignored by the server. Any subsequent part SHOULD also be considered to be an End part.

Note The server SHOULD accept and ignore a DocInfo/DocData pair if the URL in the DOCINFO is empty. Clients SHOULD avoid doing this because it wastes bandwidth.

If the "atomic" PUT-OPTION-VAL is specified as defined in Put-Option, the server SHOULD either succeed in storing all the documents or not store any of them. If the client does not request the "atomic" option, or if the server does not honor the option, and if the server is unable to store a document, then documents in the request before the one that failed MUST be stored, and documents after the one that failed MUST NOT be stored.

Tokens

service_name: This parameter is deprecated: See service_name in section 3.1.5.3.1.

put_option: A set of flags describing the client's requested behavior. For details, see Put-Option.

time_tokens: This parameter uses a VECTOR-TIME. Each time the value that is specified in the *time_tokens* parameter is checked, in addition to values specified through the vti_timelastmodified metakey and putOption = edit check. If this parameter is sent, it MUST either be empty (which is equivalent to not sending it) or it MUST contain a TIME entry for

each document that will be sent in the remainder of the request. In that case, the server SHOULD check this value in addition to the vti_timelastmodified metakey sent in each request.<31>

listFiles: A BOOLEAN value that specifies if the docs return value will be included in the response. The server MUST include the docs return value if, and only if, this parameter is true.

listLinkInfo: See section 3.1.5.3.1.

validateWelcomeNames: For semantics, see section 3.1.5.3.1.

URL

FPAuthorScriptUrl

Return Values

docs

A VECTOR-DOCINFO containing information about the saved documents. This value MUST NOT be included in the response if listFiles is passed as false.

error-index

This INT is only returned in error conditions for a single document rather than the transfer in general (for example, because of a timestamp mismatch, a document exists, or a document is checked out). If present, it MUST be the zero-based index of the document that the server was unable to store.

document

This DOC-INFO MUST be returned if, and only if, error-index is also returned. This should indicate the document URL and metainformation for the document that could not be uploaded.

3.1.5.3.13 remove documents

The remove documents request is used by the client to delete specific documents or folders from the Web site.

Tokens

service_name: This parameter is deprecated: See service_name in section 3.1.5.3.1.

url_list: A VECTOR-URL-STRING list of service-relative URLs that the client wants to be deleted. The server MUST attempt to delete the URLs listed here subject to authorization checks.

time_tokens: If present and non-empty, the value lists the vti_timelastmodified for the corresponding documents as known by the client. If the VECTOR-TIME is empty or the parameter is not present, the server MUST ignore this parameter; otherwise, it MAY refuse to delete documents in which the date does not match the actual vti_timelastmodified as known on the server.<32>

validateWelcomeNames: For semantics, see section 3.1.5.3.1.

URL

FPAuthorScriptUrl

Return Values

message

A STRING description of the action taken by the server. This is intended for debugging, and SHOULD be ignored by the client.

removed_docs

A VECTOR-DOCINFO containing the name and metainformation for the documents that were removed. The server SHOULD send empty METADICTs in this value.<33>

removed_dirs

A VECTOR-DOCINFO containing the name and metainformation for the folders that were removed. The server SHOULD send empty METADICTs in this return value.<34>

failed_docs

A VECTOR-DOCINFO specifying the name and metainformation for the documents that failed to be removed. The server MUST respond with a (potentially empty) list of documents that could not be removed.

failed_dirs

A VECTOR-URL-DIRECTORY specifying the name and metainformation for the folders that failed to be removed. The server MUST respond with a (potentially empty) list of folders that could not be removed.

3.1.5.3.14 server version

The server version request is to be used by the client to request the version of the server extensions in use on the Web site server.

Tokens

None: The argument list for this request SHOULD be empty. The server MUST ignore any parameters sent.

URL

FPShtmlScriptUrl

Return Values

server version

A VERSION specifying the current version of the server (not the effective protocol version). The server MUST respond with its actual version, which might be larger than the effective protocol version in the PROTOCOL-VERSION-STRING built in to all the FrontPage Server Extensions Remote Protocol responses, as described in section 2.2.2.3.

source control

An INT indicates if the server supports the checkout document and uncheckout document requests. This value MUST equal 0 if the server does not support these requests; otherwise, this value MUST equal 1. Nonzero values other than 1 are reserved, but the client MUST interpret any nonzero value as if it were the value 1.

As with other methods, the effective protocol version negotiated by using the mechanism defined in section 3.1.3.2 SHOULD be returned in the METHOD-KEY-VALUE element of the response.

3.1.5.3.15 uncheckout document

The uncheckout document request is used by the client to reverse a long-term checkout of a file from source control. If the file has changed since it was checked out, those changes are reverted. This request is also used to release a short-term checkout, in which case changes are not reverted. Because clients conforming to the FrontPage Server Extensions Remote Protocol MUST NOT assume the server supports long-term checkouts, only the latter use is pertinent.

Tokens

service_name: This parameter is deprecated: See service_name in section 3.1.5.3.1.

document_name: A URL-STRING specifying the Service-relative path of the current document.

force: A Boolean that reverses the checkout of a file by another user. Clients conforming to the FrontPage Server Extensions Remote Protocol MUST send this parameter as 0 (either explicitly or by not sending the parameter and taking the default). The server MAY ignore this value. If the server chooses to implement this functionality, it SHOULD do additional authorization checks and ignore the parameter if those checks fail. The value defaults to zero.<35>

time_checked_out: A TIME indicating the client's record of the time and date at which the file was last checked out. The server MAY refuse to revert a checkout if the time does not match the server's record of the time the file was checked out.

rlsshortterm: A Boolean value indicates if the client wants to release a short-term checkout or a long-term checkout. If true, the server MUST release the lock; otherwise, the server SHOULD release any long-term checkout the client has acquired. Clients conforming to the FrontPage Server Extensions Remote Protocol MUST NOT assume the server supports long-term checkout and thus MUST send true for this parameter. The server should return an appropriate error if the client does not have the kind of checkout it is trying to undo.

validateWelcomeNames: For semantics, see section 3.1.5.3.1.

URL

FPAuthorScriptUrl

Return Values

meta_info

The METADICT information about the document that has been unchecked out.

3.1.5.3.16 url to web url

The url to web url request is used by the client to decompose the URL into the server-relativeURL of the Web site that contains the URL and the server-relativeURL for the file within the Web site.

Tokens

service_name: This parameter is deprecated: See service_name in section 3.1.5.3.1.

url: A URL-STRING specifying the server-relative URL of the object that the client wants to decompose.

flags: The flags parameter. An INT MUST be ignored by the server but MAY be sent by the client and SHOULD equal zero.

URL

FPShtmlScriptUrl

Return Values

webUrl

A URL-STRING specifying the Server-relative URL of the Web site.

fileUrl

A URL-STRING specifying the Service-relative URL of the file.

3.1.5.3.17 SharePoint Team Services

SharePoint Team Services enables file sharing and discussions among members of a group. SharePoint Team Services also adds collaboration and sharing capabilities to Web sites. These capabilities allow teams to share and maintain information without the need for central administration of a Web site. SharePoint Team Services authoring and administration of Web sites complements the FrontPage Server Extensions Remote Protocol.

As with the FrontPage Server Extensions Remote Protocol, SharePoint methods are remote procedures that are transported in an HTTP request to a server. However, rather than a transport with a POST request, this SHOULD be sent as a GET or a HEAD with the standard application/x-www-form-urlencoded content type as described in HTML 2.0, [RFC1866] sections 8.2.1 and 8.2.2.

3.1.5.3.17.1 dialogview

The dialogview request is used by a client to obtain an HTML–rendered view of the document libraries within a Web site, a specific document library, or a folder within a document library, which is used in a dialog box for opening or saving files; or opens the property form that is used when saving a file.

Tokens

dialogview: Specifies the view to display. Possible values include the following:

Mask	Meaning
FileOpen	The Open dialog box. The server MUST give an HTML document suitable for rendering an open dialog; it SHOULD not encourage users to select documents that do not exist.
FileSave	The Save dialog box. The server MUST give an HTML dialog suitable for rendering a save dialog; it SHOULD allow for picking an existing document or creating a new one.
SaveForm	The Property form. To view the property form used when saving a file, the *location* parameter specifies the file in the document library. Clients conforming to the FrontPage Server Extensions Remote Protocol MUST NOT use this value for dialogview, section 3.1.5.3.17.1. The server MAY treat this as any other non-supported value.
All other	The server MUST return an HTTP 400 (GONE) response.

Mask	Meaning
values	

location: Specifies the site-relative URL of a document library or of a folder or file within a document library. If SaveForm is specified in dialogview, the URL for the *location* parameter MUST point to the file being saved. If the *location* parameter is passed without specifying a value, this method SHOULD display a view of all the document libraries in the site.

FileDialogFilterValue: Specifies the file type extension by which to filter the view in the file dialog box. For example, *.doc, *.txt, or *.htm.

URL

TPScriptURL

Return Values

Success

Displays the view of a document library or folder that is used in a file dialog box, in all document libraries in the site, or in the form used to specify document properties when saving a file.**Error**

If the server requires further authentication or authorization, it MUST trigger an appropriate response from the HTTP layer. Otherwise, if the request has the *dialogview* parameter, but has some other error (for instance, if the *location* or *dialogview* parameters are not valid) the server MUST send a 400 GONE response. The client conforming to the subset of the protocol described here MUST send the *dialogview* parameter to any request to TPScriptURL.

Comments: If the result is not an error, the server's response MUST contain an HTML table whose ID is "FileDialogView". Each file that the server wants to show the client MUST be represented as a TR element in the table that has an ID containing the server-relative URL of the file to be represented and has an expando property fileattribute="file". The server MUST do likewise for each folder except that it should have the fileattribute="folder" expando property.

If the result is an error, the HTTP return values MUST be 400 and the contents SHOULD be HTML suitable for displaying to an end user.

The client SHOULD render the HTML returned by the server, detect mouse clicks and other selection gestures that it wants to respond to, and use the HTML document object model to determine the TR on which the selection occurred. If the user selects a folder, the client SHOULD make a new dialogview request with the location specified by the folder. If the user selects a file, the client SHOULD treat that as the file the user wants to open or save over.

If the client gets an error, it MAY fall back to an alternate dialog method (potentially populated by the list documents method or present the error to the user.

3.1.6 Timer Events

3.1.6.1 Short-Term Checkout Timer Expiry

When the short-term checkout timer on a document expires, the server MUST clear the short-term checkout on the document. This will leave the document open for editing by any user. If the client wants to prevent short-term checkout from expiring, the client MUST send another checkout document request for the same document before the checkout has expired.

3.1.7 Other Local Events

None.

4 Protocol Examples

The following sections specify protocol examples.

4.1 Example Entry Point for FrontPage Server Extensions

4.1.1 First Determining the Entry Point

Each method specification gives an entry point that corresponds to one of four URLs that are returned when a client performs an HTTP GET on _vti_inf.html. This section details how to determine the URL to POST given the known entry point.

4.1.1.1 First Entry Point Example

If the client wants to call the server version method, section 3.1.5.3.14, it needs to use the FPShtmlScriptUrl entry point. For details, see section 3.1.3. If it is making this call against the root of the server, the URL is:

```
/_vti_bin/shtml.dll/_vti_rpc
```

If the client is making a call against a subsite located at /search/, the URL is:

```
/search/_vti_bin/shtml.dll/vti_rpc.
```

4.1.1.2 Second Entry Point Example

If the client wants to call the open service method, section 3.1.5.3.10, it needs to use the FPAuthorScriptURL entry point:

```
POST
/site_url/_vti_bin/_vti_aut/author.dll HTTP/1.0
.
.
.
method=open+service:6.0.n.nnnn
```

The first line shows a post to /site_url/_vti_bin/_vti_aut/author.dll, which is the FPAuthorScriptURL entry point for the subsite called 'site_url'.

4.1.2 SharePoint Services Entry Note

The TPScriptUrl field is only present on servers that have Windows SharePoint Services (or SharePoint Team Services) enabled. It is a service-relative URL and refers to the URL to POST for Windows SharePoint Services methods. For details, see section 3.1.5.3.17.

4.2 Example Trace for Posts

The following is an example trace for common operations that are performed on the client. The example shows operations such as opening a Web folder, copying and pasting to (or from) a Web folder, opening a file, saving changes in a file, and closing a file.

Note In the example, WWW-Authenticate headers, as specified in [RFC2616] section 14.47, have been removed, "DOMAIN1" is a placeholder for domain name, "testuser" is a placeholder for user name, and "fpseserver" is a placeholder for an actual server name. All the lines, except for any text files that are uploaded, should be terminated by "\n" rather than the "\r\n", which is standard on Windows operating systems.

4.2.1 Querying for URLs to Post

Any user action requiring the client to interact with the server through Microsoft FrontPage Server Extensions requires the client to know what URLs to POST. Consequently, any FrontPage Server Extensions Remote Protocol conversation will begin with the client posting an HTTP GET to /_vti_inf.html to determine the URLs of author.dll, shtml.dll (for details, see section 3.1.3.2.1), and so on.

4.2.1.1 Client HTTP GET Request for _vti_inf.html

```
GET /_vti_inf.html HTTP/1.1
Date: Thu, 08 June 2006 21:39:52 GMT
MIME-Version: 1.0
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MS FrontPage 12.0)
Host: fpseserver
Accept: auth/sicily
Content-Length: 0
Connection: Keep-Alive
Cache-Control: no-cache
```

4.2.1.2 Server HTTP Response

```
HTTP/1.1 200 OK
Content-Length: 1754
Content-Type: text/html
Last-Modified: Thu, 08 June 2006 21:04:13 GMT
Accept-Ranges: bytes
ETag: "2f7ad4cbe6fc61:33a"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
MicrosoftOfficeWebServer: 5.0_Pub
Date: Thu, 08 June 2006 21:39:42 GMT

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<title> FrontPage Configuration Information </title>
</head>

<body>
<!-- _vti_inf.html version 0.100>
<!--
This file contains important information used by the FrontPage client
(the FrontPage Explorer and FrontPage Editor) to communicate with the
FrontPage server extensions installed on this web server.
```

```
The values below are automatically set by FrontPage at installation.
Normally, you do not need to modify these values, but in case
you do, the parameters are as follows:

'FPShtmlScriptUrl', 'FPAuthorScriptUrl', and 'FPAdminScriptUrl'
specify the relative urls for the scripts that FrontPage uses for
remote authoring.  These values SHOULD NOT be changed.

'FPVersion' identifies the version of the FrontPage Server Extensions
installed, and SHOULD NOT be changed.
--><!-- FrontPage Configuration Information
    FPVersion="5.0.2.6738"
    FPShtmlScriptUrl="_vti_bin/shtml.dll/_vti_rpc"
    FPAuthorScriptUrl="_vti_bin/_vti_aut/author.dll"
    FPAdminScriptUrl="_vti_bin/_vti_adm/admin.dll"
    TPScriptUrl="_vti_bin/owssvr.dll"
-->
<p><!--webbot bot="PurpleText"
preview="This page is placed into the root directory of your FrontPage
web when FrontPage is installed.  It contains information used by the
FrontPage client to communicate with the FrontPage Server Extensions
installed on this web server.  You SHOULD NOT delete this file."
--></p>

<h1>FrontPage Configuration Information </h1>

<p>In the HTML comments, this page contains configuration
information that the FrontPage Explorer and FrontPage Editor need to
communicate with the FrontPage Server Extensions installed on
this web server. Do not delete this page.</p>
</body>
</html>
```

4.2.2 Opening a Web Folder

This example uses the server version request, section 3.1.5.3.14, to query the server's version and uses the list documents request, section 3.1.5.3.8, to enumerate the documents in the root of the server. This part of the example corresponds to opening a folder as a Web folder in a Web browser.

4.2.2.1 Client Calls server version Method

```
Date: Thu, 08 June 2006 21:39:52 GMT
MIME-Version: 1.0
User-Agent: MSFrontPage/12.0
Host: fpseserver
Accept: auth/sicily
Content-Length: 42
Content-Type: application/x-www-form-urlencoded
X-Vermeer-Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cache-Control: no-cache

method=server+version%3a12%2e0%2e0%2e3417
```

4.2.2.2 Server Responds to server version Method

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 08 June 2006 21:39:42 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
MicrosoftOfficeWebServer: 5.0_Pub
Content-type: application/x-vermeer-rpc

<html><head><title>vermeer RPC packet</title></head>
<body>
<p>method=server version:5.0.2.6738
<p>server version=
<list>
<item>major ver=5
<item>minor ver=0
<item>phase ver=2
<item>ver incr=6738
</list>
<p>source control=1
</body>
</html>
```

4.2.2.3 Client Calls list documents Method

```
POST /_vti_bin/_vti_aut/author.dll HTTP/1.1
Date: Thu, 08 June 2006 21:40:01 GMT
MIME-Version: 1.0
User-Agent: MSFrontPage/12.0
Host: fpseserver
Accept: auth/sicily
Content-Length: 336
Content-Type: application/x-www-form-urlencoded
X-Vermeer-Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cache-Control: no-cache

method=list+documents%3a5%2e0%2e2%2e6738&service%5fname=
&listHiddenDocs=false&listExplorerDocs=false&listRecurse=
false&listFiles=true&listFolders=true&listLinkInfo=
false&listIncludeParent=true&listDerived=false&listBorders=
false&listChildWebs=true&listThickets=true&initialUrl=&folderList=
%5b%3bTW%7c08+June+2006+21%3a04%3a14+%2d0000%5d
```

4.2.2.4 Server Responds to list documents Method

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 08 June 2006 21:39:51 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
MicrosoftOfficeWebServer: 5.0_Pub
Content-type: application/x-vermeer-rpc
```

```
<html><head><title>vermeer RPC packet</title></head>
<body>
<p>method=list documents:5.0.2.6738
<p>document_list=
<list>
<list>
<item>document_name=Thicket test.htm
<item>meta_info=
<list>
<item>vti_author
<item>SR|DOMAIN1&#92;testuser
<item>vti_modifiedby
<item>SR|DOMAIN1&#92;testuser
<item>vti_timelastmodified
<item>TR|08 June 2006 21:28:52 -0000
<item>vti_timecreated
<item>TR|08 June 2006 21:27:31 -0000
<item>vti_title
<item>SW|Test
<item>vti_nexttolasttimemodified
<item>TR|08 June 2006 21:28:39 -0000
<item>vti_filesize
<item>IR|930
<item>vti_metatags
<item>VR|HTTP-EQUIV&#61;Content-Type text/html&#59;&#92;charset&#61;
windows-1252 Generator Microsoft&#92; Word&#92; 12&#92; (filtered)
<item>vti_charset
<item>SR|windows-1252
<item>vti_generator
<item>SR|Microsoft Word 12 (filtered)
<item>vti_timelastwritten
<item>TX|08 June 2006 21:28:52 -0000
</list>
</list>
</list>
<p>urldirs=
<list>
<list>
<item>url=
<item>meta_info=
<list>
<item>vti_isexecutable
<item>BR|false
<item>vti_isbrowsable
<item>BR|true
<item>vti_isscriptable
<item>BR|true
<item>vti_hassubdirs
<item>BR|true
<item>vti_dirlateststamp
<item>TW|08 June 2006 21:28:52 -0000
</list>
</list>
<list>
<item>url=aspnet_client
<item>meta_info=
<list>
<item>vti_isexecutable
```

```
<item>BR|false
<item>vti_isbrowsable
<item>BR|true
<item>vti_isscriptable
<item>BR|false
<item>vti_hassubdirs
<item>BR|true
</list>
</list>
<list>
<item>url=images
<item>meta_info=
<list>
<item>vti_isexecutable
<item>BR|false
<item>vti_isbrowsable
<item>BR|true
<item>vti_isscriptable
<item>BR|true
<item>vti_hassubdirs
<item>BR|false
</list>
</list>
<list>
<item>url=Thicket Test_files
<item>meta_info=
<list>
<item>vti_isexecutable
<item>BR|false
<item>vti_isbrowsable
<item>BR|true
<item>vti_isscriptable
<item>BR|true
<item>vti_hassubdirs
<item>BR|false
</list>
</list>
<list>
<item>url=_private
<item>meta_info=
<list>
<item>vti_isexecutable
<item>BR|false
<item>vti_isbrowsable
<item>BR|false
<item>vti_isscriptable
<item>BR|false
<item>vti_hassubdirs
<item>BR|false
</list>
</list>
</list>
</body>
</html>
```

4.2.3 Copying a File to a Web Folder

This example uses the url to web url request, section 3.1.5.3.16, to discover where a file (in this case, /small.txt) belongs, and uses the put document request, section 3.1.5.3.11, to upload it. This part of the example corresponds to a copy/paste operation into the Web folder.

4.2.3.1 Client Calls url to web url Method

```
POST /_vti_bin/shtml.dll/_vti_rpc HTTP/1.1
Date: Thu, 08 June 2006 21:40:17 GMT
MIME-Version: 1.0
User-Agent: MSFrontPage/12.0
Host: fpseserver
Accept: auth/sicily
Content-Length: 68
Content-Type: application/x-www-form-urlencoded
X-Vermeer-Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cache-Control: no-cache

method=url+to+web+url%3a5%2e0%2e2%2e6738&url=%2fsmall%2etxt&flags=0
```

4.2.3.2 Server Responds to url to web url Method

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 08 June 2006 21:40:07 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
MicrosoftOfficeWebServer: 5.0_Pub
Content-type: application/x-vermeer-rpc

<html><head><title>vermeer RPC packet</title></head>
<body>
<p>method=url to web url:5.0.2.6738
<p>webUrl=/
<p>fileUrl=small.txt
</body>
</html>
```

4.2.3.3 Client Calls put document Method

```
POST /_vti_bin/_vti_aut/author.dll HTTP/1.1
Date: Thu, 08 June 2006 21:40:17 GMT
MIME-Version: 1.0
User-Agent: MSFrontPage/12.0
Host: fpseserver
Accept: auth/sicily
Content-Length: 224
Content-Type: application/x-vermeer-urlencoded
X-Vermeer-Content-Type: application/x-vermeer-urlencoded
Connection: Keep-Alive
Cache-Control: no-cache

method=put+document%3a5%2e0%2e2%2e6738&service%5fname=&document=%5b
```

```
document%5fname%3dsmall%2etxt%3bmeta%5finfo%3d%5b%5d%5d&put%5foption
=edit%2catomic%2cthicket&comment=&keep%5fchecked%5fout=false
This is a small text file.
```

4.2.3.4 Server Responds to put document Method

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 08 June 2006 21:40:07 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
MicrosoftOfficeWebServer: 5.0_Pub
Content-type: application/x-vermeer-rpc

<html><head><title>vermeer RPC packet</title></head>
<body>
<p>method=put document:5.0.2.6738
<p>message=successfully put document 'small.txt' as 'small.txt'
<p>document=
<list>
<item>document_name=small.txt
<item>meta_info=
<list>
<item>vti_author
<item>SR|DOMAIN1&#92;testuser
<item>vti_modifiedby
<item>SR|DOMAIN1&#92;testuser
<item>vti_timelastmodified
<item>TR|08 June 2006 21:40:07 -0000
<item>vti_timecreated
<item>TR|08 June 2006 21:40:07 -0000
<item>vti_filesize
<item>IR|28
<item>vti_backlinkinfo
<item>VX|
<item>vti_timelastwritten
<item>TX|08 June 2006 21:40:07 -0000
</list>
</list>
</body>
</html>
```

4.2.4 Downloading a File from a Web Folder

This example uses the get document request, section 3.1.5.3.6, to download the file. This part of the example corresponds to a copy/paste operation from the Web folder.

4.2.4.1 Client Calls get document Method

```
POST /_vti_bin/_vti_aut/author.dll HTTP/1.1
Date: Thu, 08 June 2006 21:40:30 GMT
MIME-Version: 1.0
User-Agent: MSFrontPage/12.0
Host: fpseserver
Accept: auth/sicily
```

```
Content-Length: 162
Content-Type: application/x-www-form-urlencoded
X-Vermeer-Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cache-Control: no-cache

method=get+document%3a5%2e0%2e2%2e6738&service%5fname=&document
%5fname=small%2etxt&old%5ftheme%5fhtml=false&force=true&get
%5foption=none&doc%5fversion=&timeout=0
```

4.2.4.2 Server Responds to get document Method

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 08 June 2006 21:40:20 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
MicrosoftOfficeWebServer: 5.0_Pub
Content-type: application/x-vermeer-rpc

<html><head><title>vermeer RPC packet</title></head>
<body>
<p>method=get document:5.0.2.6738
<p>message=successfully retrieved document 'small.txt' from
'small.txt'
<p>document=
<list>
<item>document_name=small.txt
<item>meta_info=
<list>
<item>vti_author
<item>SR|DOMAIN1&#92;testuser
<item>vti_modifiedby
<item>SR|DOMAIN1&#92;testuser
<item>vti_timelastmodified
<item>TR|08 June 2006 21:40:07 -0000
<item>vti_timecreated
<item>TR|08 June 2006 21:40:07 -0000
<item>vti_filesize
<item>IR|28
<item>vti_backlinkinfo
<item>VX|
<item>vti_timelastwritten
<item>TX|08 June 2006 21:40:07 -0000
</list>
</list>
</body>
</html>
This is a small text file.
```

4.2.5 Opening a File in a Web Folder

When opening a file, as seen in the next part of the example, a client application calls the get document request, section 3.1.5.3.6, with a time-out of a 10-minute short-term checkout, as can be

seen at the end of the get document request, in the section below. This guarantees that the document cannot be modified by other users while it is open in the client application.

4.2.5.1 Client Calls get document Method

```
POST /_vti_bin/_vti_aut/author.dll HTTP/1.1
Date: Thu, 08 June 2006 21:41:45 GMT
MIME-Version: 1.0
User-Agent: MSFrontPage/12.0
Host: fpseserver
Accept: auth/sicily
Content-Length: 175
Content-Type: application/x-www-form-urlencoded
X-Vermeer-Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cache-Control: no-cache

method=get+document%3a5%2e0%2e2%2e6738&service%5fname=&document%5f
name=small%2etxt&old%5ftheme%5fhtml=false&force=false&get%5foption=
chkoutExclusive&doc%5fversion=&timeout=10
```

4.2.5.2 Server Responds to get document Method

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 08 June 2006 21:41:35 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
MicrosoftOfficeWebServer: 5.0_Pub
Content-type: application/x-vermeer-rpc

<html><head><title>vermeer RPC packet</title></head>
<body>
<p>method=get document:5.0.2.6738
<p>message=successfully retrieved document 'small.txt' from
'small.txt'
<p>document=
<list>
<item>document_name=small.txt
<item>meta_info=
<list>
<item>vti_author
<item>SR|DOMAIN1&#92;testuser
<item>vti_modifiedby
<item>SR|DOMAIN1&#92;testuser
<item>vti_timelastmodified
<item>TR|08 June 2006 21:40:07 -0000
<item>vti_timecreated
<item>TR|08 June 2006 21:40:07 -0000
<item>vti_filesize
<item>IR|28
<item>vti_backlinkinfo
<item>VX|
<item>vti_sourcecontrollockexpires
<item>TR|08 June 2006 21:51:35 -0000
<item>vti_sourcecontrolcheckedoutby
```

```
<item>SR|DOMAIN1&#92;testuser
<item>vti_sourcecontrolmultiuserchkoutby
<item>VR|DOMAIN1&#92;&#92;testuser
<item>vti_timelastwritten
<item>TX|08 June 2006 21:40:07 -0000
</list>
</list>
</body>
</html>
This is a small text file.
```

4.2.6 Saving a File to a Web Folder

Changing and saving a file, as seen in the next part of the example, requires calling the put document request, section 3.1.5.3.11.

4.2.6.1 Client Calls put document Method

```
POST /_vti_bin/_vti_aut/author.dll HTTP/1.1
Date: Thu, 08 June 2006 21:41:57 GMT
MIME-Version: 1.0
User-Agent: MSFrontPage/12.0
Host: fpseserver
Accept: auth/sicily
Content-Length: 290
Content-Type: application/x-vermeer-urlencoded
X-Vermeer-Content-Type: application/x-vermeer-urlencoded
Connection: Keep-Alive
Cache-Control: no-cache

method=put+document%3a5%2e0%2e2%2e6738&service%5fname=&document=%5
bdocument%5fname%3dsmall%2etxt%3bmeta%5finfo%3d%5bvti%5ftimelastmodi
fied%3bTW%7c08+June+2006+21%3a40%3a07+%2d0000%5d%5d&put%5foption=edi
t&comment=&keep%5fchecked%5fout=false
This is a small text file. Now, a little bigger.
```

4.2.6.2 Server Responds to put document Method

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 08 June 2006 21:41:47 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
MicrosoftOfficeWebServer: 5.0_Pub
Content-type: application/x-vermeer-rpc

<html><head><title>vermeer RPC packet</title></head>
<body>
<p>method=put document:5.0.2.6738
<p>message=successfully put document 'small.txt' as 'small.txt'
<p>document=
<list>
<item>document_name=small.txt
<item>meta_info=
<list>
```

```
<item>vti_author
<item>SR|DOMAIN1&#92;testuser
<item>vti_modifiedby
<item>SR|DOMAIN1&#92;testuser
<item>vti_timelastmodified
<item>TR|08 June 2006 21:41:47 -0000
<item>vti_timecreated
<item>TR|08 June 2006 21:40:07 -0000
<item>vti_backlinkinfo
<item>VX|
<item>vti_sourcecontrollockexpires
<item>TR|08 June 2006 21:51:35 -0000
<item>vti_sourcecontrolcheckedoutby
<item>SR|DOMAIN1&#92;testuser
<item>vti_sourcecontrolmultiuserchkoutby
<item>VR|DOMAIN1&#92;&#92;testuser
<item>vti_nexttolasttimemodified
<item>TW|08 June 2006 21:40:07 -0000
<item>vti_filesize
<item>IR|51
<item>vti_timelastwritten
<item>TX|08 June 2006 21:41:47 -0000
</list>
</list>
</body>
</html>
```

4.2.7 Closing a File

Finally, this example shows what happens when the file is closed in the client application, which requires a call to the uncheckout document request, section 3.1.5.3.15, to release the lock. Note that the example does not illustrate the effects of waiting 10 minutes to cause the client application to renew the short-term checkout, which would have caused a checkout document request, section 3.1.5.3.2, to be sent with a *timeout* parameter.

4.2.7.1 Calls uncheckout document Method

```
POST /_vti_bin/_vti_aut/author.dll HTTP/1.1
Date: Thu, 08 June 2006 21:41:59 GMT
MIME-Version: 1.0
User-Agent: MSFrontPage/12.0
Host: fpseserver
Accept: auth/sicily
Content-Length: 120
Content-Type: application/x-www-form-urlencoded
X-Vermeer-Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Cache-Control: no-cache

method=uncheckout+document%3a5%2e0%2e2%2e6738&service%5fname=
&document%5fname=small%2etxt&force=false&rlsshortterm=true
```

4.2.7.2 Server Responds to uncheckout document Method

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 08 June 2006 21:41:49 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
MicrosoftOfficeWebServer: 5.0_Pub
Content-type: application/x-vermeer-rpc

<html><head><title>vermeer RPC packet</title></head>
<body>
<p>method=uncheckout document:5.0.2.6738
<p>meta_info=
<list>
<item>vti_author
<item>SR|DOMAIN1&#92;testuser
<item>vti_modifiedby
<item>SR|DOMAIN1&#92;testuser
<item>vti_timelastmodified
<item>TR|08 June 2006 21:41:47 -0000
<item>vti_timecreated
<item>TR|08 June 2006 21:40:07 -0000
<item>vti_backlinkinfo
<item>VX|
<item>vti_nexttolasttimemodified
<item>TW|08 June 2006 21:40:07 -0000
<item>vti_filesize
<item>IR|51
<item>vti_timelastwritten
<item>TX|08 June 2006 21:41:47 -0000
</list>
</body>
</html>
```

5 Security

The following sections specify the security considerations for implementers.

5.1 Security Considerations for Implementers

5.1.1 One-Click Attacks

It is possible for an attacker to lure a user to a malicious page, such as by sending the user a URL in e-mail. When the user visits the malicious page, that page can perform a silent POST to the server. Because the FontPage Server Extensions Remote Protocol is merely an HTTP POST, this means the attacker can lure the user into performing any FrontPage Server Exensions Remote Protocol operation against any server. This sort of attack is termed a one-click attack.

To prevent this type of attack, servers should require all incoming FrontPage Server Extensions Remote Protocol requests to have the HTTP header X-Vermeer-Content-Type, as specified in [RFC2616] section 14.17. Because normal Web browsers do not send this header, requiring it effectively prevents users from browsing to a page that can execute a silent FrontPage Server Extensions Remote Protocol method call. It is strongly recommended that all implementations of the FrontPage Server Extensions Remote Protocol require this header to prevent one-click attacks.

5.1.2 Permissions for Entry Points

Servers have traditionally restricted access to methods to certain classes of users. Although this restriction is not required by the FrontPage Server Extensions Remote Protocol, it is recommended because some methods, such as remove documents, section 3.1.5.3.13, can be damaging to user data.

The FrontPage Server Extensions Remote Protocol has traditionally determined which users can call which methods based on the method entry points. Methods whose entry point is FPShtmlScriptUrl can usually be called by any user. Methods with the FPAuthorScriptURL entry point are restricted to users who can read or write documents on the server. The reason for this model is that methods such as remove documents, section 3.1.5.3.13, are considered more dangerous than server version, section 3.1.5.3.14. As such, restricting unauthenticated users from even calling the more powerful methods provides an extra layer of security.

Implementers of the FrontPage Server Extensions Remote Protocol are free to restrict method entry point security if they choose to, or they can rely on the object permissions discussed below.

5.1.3 Permissions for Objects

Like most file systems, FrontPage Server Extensions Remote Protocol objects can have a notion of security. The granularity of this security is up to the server implementers. Windows implementations of the FrontPage Server Extensions Remote Protocol provide the capability to granularly control read access and write access on files, folders, and services. On a secured server, each method call should check the appropriate rights before executing. If the user does not have sufficient rights, the implementation should trigger the HTTP layer to return a 401 message, access denied. The HTTP layer on the client and server should then manage authenticating the user, if that user does in fact have permissions.

5.2 Index of Security Parameters

None.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows NT
- Windows 2000
- Windows XP
- Windows Server 2003
- Windows Vista

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

<1> Section 1: Windows Vista only supports the SharePoint Team Services dialogview aspect of the FrontPage Server Extensions Remote Protocol. All other aspects of the protocol are deprecated in favor of WebDAV. For more information about WebDAV, see [MS-WDV].

<2> Section 2.2.2.5.12: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 ignore this parameter; versions 4.0 and 5.0 do not.

<3> Section 2.2.2.5.12: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 do not ignore this parameter; all other versions ignore it.

<4> Section 2.2.2.5.12: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 accept this parameter and require the requesting user to be a Web administrator.

<5> Section 2.2.2.5.12: The FrontPage Server Extensions Remote Protocol versions 4.0 and 5.0 accept this parameter if source control is enabled.

<6> Section 2.2.2.5.13: The FrontPage Server Extensions:Website Management protocol versions 4.0 and 5.0 accept this parameter.

<7> Section 2.2.4.12: FrontPage Server Extensions Remote Protocol Server versions 4.0 and 5.0 set a value here that reflects the underlying file system it stores documents in. Versions 6.0 and 12.0 always send 1 for this value.

<8> Section 2.2.4.13: The Windows client uses this metadata to avoid fetching the content of the file just to discover metatags with NAME="progid" and NAME="generator"; these are used to display icons for HTML files and to select an appropriate editor.

<9> Section 2.2.4.15: Windows servers use a string in the form DOMAIN\user name to identify a user.

<10> Section 2.2.4.20: The FrontPage Server Extensions Remote Protocol client for versions 4.0, 5.0, 6.0, and 12.0 uses this value when rendering a file's or folder's time last modified.

<11> Section 2.2.4.21: The FrontPage Server Extensions Remote Protocol Server versions 4.0, 5.0, and 6.0 do not include this key. The FrontPage Server Extensions Remote Protocol Client versions 4.0, 5.0, and 6.0 are not sensitive to this key.

<12> Section 2.2.4.23: FrontPage Server Extensions Remote Protocol Server versions 4.0 and 5.0 do not include this key, versions 6.0 and 12.0 do. FrontPage Server Extensions Remote Protocol Client versions 4.0 and 5.0 only consider information passed to the underlying HTTP authentication mechanism.

<13> Section 3.1.1.1: The FrontPage Server Extensions Remote Protocol versions 4.0 and 5.0 allow users to turn the source control sandbox off; versions 6.0 and 12.0 do not.

<14> Section 3.1.2.1: All Windows operating systems request a short-term checkout length of two minutes. The clients will attempt to renew the short-term checkout 10 seconds before it expires.

<15> Section 3.1.3.2.1: Windows Vista does not perform this GET, and instead assumes the values shown in the example in section 3.1.3.2.1.

<16> Section 3.1.3.2.2: All FrontPage Server Extensions Remote Protocol server versions return the URsL listed.

<17> Section 3.1.5.1: The FrontPage Server Extensions Remote Protocol versions 4.0 and 5.0 do not fail the request and instead use the normal HTTP Content-Type header. This has potential security implications (see section 5.1).

<18> Section 3.1.5.1: If the client does not include FrontPage in its User-Agent string, all versions of Windows will respond with the HTTP Content-Type as "text/html" and present more simplistic error strings.

<19> Section 3.1.5.2: Versions 4.0, 5.0, 6.0, and 12.0 of the FrontPage Server Extensions Remote Protocol server will treat unknown arguments as a syntax error if the method takes any parameters. For methods that take no parameters, such as server version (section 3.1.5.3.14), the FrontPage Server Extensions Remote Protocol server will ignore the parameters.

<20> Section 3.1.5.2: The FrontPage Server Extensions Remote Protocol server versions 5.0, 6.0, and 12.0 will erroneously return a badly-formed response message body which is not compliant with [RFC2616] for method calls made without authentication that result in an HTTP 401 error response. FrontPage Server Extensions Remote Protocol server version 4.0 does not have this defect and does not emit a response message body with an HTTP 401 error response.

All of the methods listed above in section 3.1.5.3 are known to have the defect in FrontPage Server Extensions Remote Protocol server versions 5.0, 6.0, and 12.0 when returning an HTTP 401 response, with the exceptions of get documents, server version, url to web url, and dialogview.

This is an example of the badly-formed message body returned by version 5.0 of FrontPage Server Extensions from a typical method call made without authentication, in this case, "open service":

```
<html dir="ltr">
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
name="CharsetDefinition">
</HEAD><body ID=idErr><p><H2>You are not authorized to view this page</H2></p>

<p>You do not have permission to view this page using your current user account.<br>
Please try the following:<br>
<li>If you have another user account with a higher level of permission, click <br>
your browser's Back button to try again using that account.
</li><li>If you believe you should be able to view this page, contact the Web site
administrator.</li></p>

</body></html>
<html><head><title>vermeer RPC packet</title></head>
<body>
```

```
<p>method=open service:5.0.2.6738
<p>status=
<ul>
<li>status=917505
<li>osstatus=0
<li>msg=The user 'DOMAIN&#92;username' is not authorized to execute the 'Author Pages'
method.
<li>osmsg=
</ul>
</body>
</html>
```

This is an example of the badly-formed message body returned by version 12.0 of the FrontPage Server Extensions Remote Protocol; the result from version 6.0 is identical save for the version number reported in the method=open service line:

```
<html dir="ltr">
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
name="CharsetDefinition">
</HEAD><body ID=idErr><p><H2>Access denied.</H2></p>

<p>You do not have permission to perform this action or access this resource.</p>

<!-- commentElt Access denied. --></body></html>
<html><head><title>vermeer RPC packet</title></head>
<body>
<p>method=open service:12.0.0.4518
<p>status=
<ul>
<li>status=917556
<li>osstatus=0
<li>msg=You are not authorized to execute this operation.
<li>osmsg=
</ul>
</body>
</html>
```

The response message body created by the FrontPage Server Extensions Remote Protocol server software exhibiting this defect is badly-formed due to the presence of two separate <HTML> sections, which may cause unexpected behavior in an insufficiently robust client that attempts to render or otherwise make use of the body.

All existing FrontPage Server Extensions Remote Protocol clients ignore the message body, if any, returned with an HTTP 401 response. Since an update or future version of FrontPage Server Extensions Remote Protocol server may correct this defect, clients MUST ignore the message body.

<21> Section 3.1.5.3: The information for these requests applies to server extensions for versions of Microsoft FrontPage 2000 and later.

<22> Section 3.1.5.3.1: The FrontPage Server Extensions Remote Protocol versions 4.0, 5.0, 6.0, and 12.0 (all versions) send this parameter.

<23> Section 3.1.5.3.1: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 ignore this parameter. Versions 4.0 and 5.0 add a new rpc_stats return value with information about how much time the command spent running, waiting, and so on. This information is only useful for debugging purposes and should not be considered part of the protocol.

<24> Section 3.1.5.3.4: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 do not have the notion of an executable directory. Versions 4.0 and 5.0 create a Web server directory with execution set to "Scripts and Executables".

<25> Section 3.1.5.3.6: Servers running the FrontPage Server Extensions Remote Protocol versions 4.0 and 5.0 return an error if chkoutNonExclusive is passed in the get_option parameter.

<26> Section 3.1.5.3.8: The FrontPage Server Extensions Remote Protocol clients send listDerived=false in the request and do not request the contents of a _derived folder.

<27> Section 3.1.5.3.8: The FrontPage Server Extensions Remote Protocol clients MUST not send this value and servers MAY ignore this value if received.

<28> Section 3.1.5.3.8: The FrontPage Server Extensions Remote Protocol versions 4.0 and 5.0 enumerate a folder (not accessible in the URL namespace) that this identifies. The default configuration has folders named "WinI386" and "all". Server administrators can install modules into these folders or create new sibling directories.

<29> Section 3.1.5.3.8: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 servers return an empty bot_list.

<30> Section 3.1.5.3.10: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 ignore this parameter.

<31> Section 3.1.5.3.12: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 do not support this behavior; however, versions 4.0 and 5.0 do.

<32> Section 3.1.5.3.13: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 do not support this behavior; however, versions 4.0 and 5.0 do.

<33> Section 3.1.5.3.13: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 send an empty METADICTs, and versions 4.0 and 5.0 send the METADICT that existed before the document was deleted.

<34> Section 3.1.5.3.13: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 send an empty METADICTs, and versions 4.0 and 5.0 send the METADICT that existed before the document was deleted.

<35> Section 3.1.5.3.15: The FrontPage Server Extensions Remote Protocol versions 6.0 and 12.0 require a special break checkout right; versions 4.0 and 5.0 ignore this parameter.

7 Index

U

V

W