[MS-VERSS]:

Versions Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.

- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.

- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.

- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability
6/27/2008	1.0	Major	Revised and edited the technical content
10/6/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
7/13/2009	1.03	Major	Revised and edited the technical content
8/28/2009	1.04	Editorial	Revised and edited the technical content
11/6/2009	1.05	Editorial	Revised and edited the technical content
2/19/2010	2.0	Editorial	Revised and edited the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Editorial	Revised and edited the technical content
6/29/2010	2.04	Minor	Clarified the meaning of the technical content.
7/23/2010	2.05	Editorial	Changed language and formatting in the technical content.
9/27/2010	2.05	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.05	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.05	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.05	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.0	Major	Significantly changed the technical content.
4/11/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	3.1	Minor	Clarified the meaning of the technical content.
9/12/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	3.2	Minor	Clarified the meaning of the technical content.
2/11/2013	3.3	Minor	Clarified the meaning of the technical content.
7/30/2013	3.3	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	3.3	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
2/10/2014	3.3	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	3.3	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	3.3	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	3.3	None	No changes to the meaning, language, or formatting of the technical content.
3/16/2015	4.0	Major	Significantly changed the technical content.
2/26/2016	5.0	Major	Significantly changed the technical content.
7/15/2016	5.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	5.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1   Introduction

This Versions Web Service Protocol enables a protocol client to view, delete, and restore a specified version of an existing file on a protocol server.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1   Glossary

This document uses the following terms:

check in: The process of placing a file or project into a source repository. This releases the lock for editing and enables other users to view the updated file or check out the file. See also check out.

checked out: A **publishing level** that indicates that a document has been created and locked for exclusive editing by a user in a version control system.

current version: The latest version of a document that is available to a user, based on the permissions of the user and the publishing level of the document.

document library: A type of list that is a container for documents and folders.

file: A single, discrete unit of content.

folder: A file system construct. File systems organize a volume's data by providing a hierarchy of objects, which are referred to as folders or directories, that contain files and can also contain other folders.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

list: A container within a SharePoint site that stores list items. A list has a customizable schema that is composed of one or more fields.

major version: An iteration of a software component, document, or list item that is ready for a larger group to see, or has changed significantly from the previous major version. For an item on a SharePoint site, the **minor version** is always "0" (zero) for a major version.

minor version: An iteration of a software component, document, or list item that is in progress or has changed only slightly from the previous version. For an item on a SharePoint site, the minor version number is never "0" (zero) and is incremented for each new version of an item, unless a **major version** is explicitly published. When minor versioning is disabled on a SharePoint site, only major version numbers are incremented, and the minor version is always "0" (zero).

published version: The version of a list item that is approved and can be seen by all users. The user interface (UI) version number for a published version is incremented to the next positive major version number and the minor version is "0" (zero). See also **major version** and **minor version**.

publishing level: An integer that is assigned to a document to indicate the publishing status of that version of the document.

Recycle Bin: The location where deleted files are stored until they are either restored, if they were deleted erroneously, or destroyed permanently.

site: A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and web site.

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

SOAP action: The HTTP request header field used to indicate the intent of the **SOAP** request, using a URI value. See [SOAP1.1] section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a SOAP message to its recipient. See [SOAP1.2-1/2007] section 5.3 for more information.

SOAP fault: A container for error and status information within a SOAP message. See [SOAP1.2-1/2007] section 5.4 for more information.

SOAP operation: An action that can be performed by a **Simple Object Access Protocol (SOAP)** service, as described in [SOAP1.1].

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

WSDL operation: A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [XML].

XML node: The smallest unit of a valid, complete structure in an XML document. For example, a node can represent an element, an attribute (1), or a text string.

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition refers to the fully specified and currently recommended standard for use in authoring XML schemas.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

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.

[ISO-8601] International Organization for Standardization, "Data Elements and Interchange Formats - Information Interchange - Representation of Dates and Times", ISO/IEC 8601:2004, December 2004, http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874&ICS1=1&ICS2=1 40&ICS3=30

Note There is a charge to download the specification.

[MS-LISTSWS] Microsoft Corporation, "Lists Web Service Protocol".

[MS-WSSTS] Microsoft Corporation, "Windows SharePoint Services".

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

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

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[SOAP1.2/1] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, http://www.w3.org/TR/2003/REC-soap12-part1-20030624

[SOAP1.2/2] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 2: Adjuncts", W3C Recommendation, June 2003, http://www.w3.org/TR/2003/REC-soap12-part2-20030624

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 Informative References

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

1.3 Protocol Overview (Synopsis)

This protocol enables programmatic access to view and manage the versions of **files** on the protocol server. It enables the retrieval of all versions of a file, the deletion of previous versions of a file, and the restoration of a file to a previous version. File versions are only generated when versioning is enabled.

Each method in the protocol is a **SOAP operation** that accepts a set of parameters as a **SOAP** request and returns a set of values as a SOAP response.

The protocol client sends a request to the protocol server via a SOAP request message, and the protocol server sends return values to the protocol client via a SOAP response message, as shown in the following figure. All SOAP requests are made to one of several well-defined **URLs** on the protocol server, which protocol clients can discover. The protocol server never initiates any communication with the protocol client.
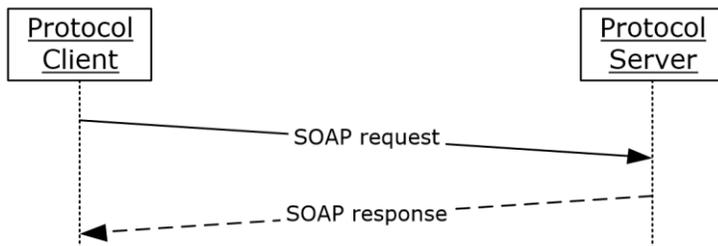
Figure 1: SOAP message sequence

1.4 Relationship to Other Protocols

This protocol uses the SOAP message protocol for formatting request and response messages, as described in [SOAP1.1], [SOAP1.2/1] and [SOAP1.2/2]. It transmits those messages by using HTTP, as described in [RFC2616], or Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS), as described in [RFC2818].

The Versions Web Service Protocol uses SOAP over HTTP, as described in [RFC2616], and SOAP over HTTPS, as described in [RFC2818], as shown in the following layering diagram.
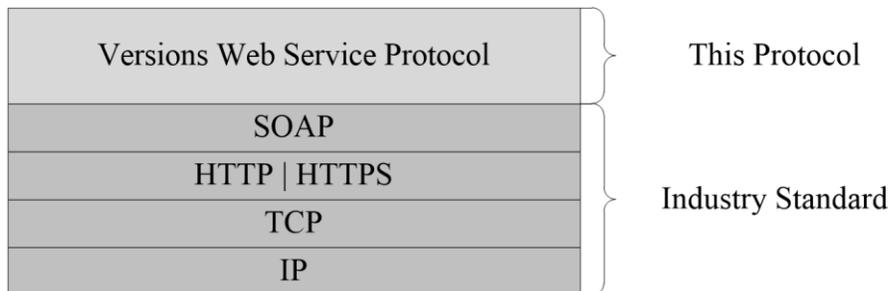
Figure 2: This protocol in relation to other protocols

This protocol calls the **CheckOutFile**, **CheckInFile**, and **UndoCheckOut** operations, as described in [MS-LISTSWS].

1.5   Prerequisites/Preconditions

This protocol operates against a **site** that is identified by a URL that protocol clients recognize. The protocol server endpoint is formed by appending "/_vti_bin/versions.asmx" to the URL of the site. Here is an example of a protocol server endpoint:

```
http://www.contoso.com/Repository/_vti_bin/versions.asmx
```

This protocol assumes that the underlying protocols have performed authentication.

1.6   Applicability Statement

This protocol is applicable in the following scenarios:

- Retrieving information about all versions of a file on the protocol server

- Deleting previous versions of a file on the protocol server

- Restoring a file on the protocol server to a previous version

1.7   Versioning and Capability Negotiation

This protocol uses multiple transports with SOAP, as specified in section 2.1.

1.8   Vendor-Extensible Fields

None.

1.9   Standards Assignments

None.

2   Messages

2.1   Transport

Protocol servers MUST support SOAP over HTTP. Protocol servers SHOULD additionally support SOAP over HTTPS for enhancing the security of communication with protocol clients.

Protocol messages MUST be formatted as specified in either [SOAP1.1] section 4 or [SOAP1.2/1] section 5. Protocol server faults MUST be returned via HTTP status codes, as specified in [RFC2616] section 10, or via **SOAP faults**, as specified in either [SOAP1.1] section 4.4 or [SOAP1.2/1] section 5.4.

2.2   Common Message Syntax

This section contains common definitions that this protocol uses. The syntax of the definitions uses XML schema as defined in [XMLSCHEMA1] and [XMLSCHEMA2], and **WSDL** as defined in [WSDL].

2.2.1   Namespaces

This specification defines and references **XML namespaces** by using the mechanisms specified in [XMLNS]. Although this specification associates a specific **XML namespace prefix** for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
Http	http://schemas.xmlsoap.org/wsdl/http/	
Mime	http://schemas.xmlsoap.org/wsdl/mime/	
S	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
Soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
Soapenc	http://schemas.xmlsoap.org/soap/encoding/	
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
Tm	http://microsoft.com/wsdl/mime/textMatching/	
Tns	http://schemas.microsoft.com/sharepoint/soap/	
Wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
(none)	http://schemas.microsoft.com/sharepoint/soap/	

2.2.2   Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

The following table summarizes the set of common XML schema element definitions defined by this specification. XML schema element definitions that are specific to a particular operation are described with the operation.

Element	Description
fileName	Defines the location of a file on the protocol server.
fileVersion	Defines the version number of the file on the protocol server.

2.2.3.1 fileName

The **fileName** element defines the location of a file on the protocol server.

<s:element name="fileName" type="s:string" />

The **fileName** element MUST be specified in one of the following formats:

- The URL of the file

- The location of the file relative to the site that contains it

The **fileName** element MUST be specified in the form *folder1*/*folder2*/…/*file-name*. The placeholders are defined as follows:

- The *folder1* and *folder2* placeholders represent the names of **folders**; *folder1* represents the topmost folder on the site.

- The *file-name* placeholder represents the name of the file.

2.2.3.2 fileVersion

The **fileVersion** element defines the version number of a file on the protocol server. The **fileVersion** element SHOULD contain the **major version** and **minor version** numbers connected by period, for example, "1.0".

```
<s:element name="fileVersion" type="s:string" />
```

2.2.4 Complex Types

The following table summarizes the set of common XML schema complex type definitions defined by this specification. XML schema complex type definitions that are specific to a particular operation are described with the operation.

Complex type	Description
Results	The details about all versions of a file on the protocol server.

Complex type	Description
SOAPFaultDetails	The details about a SOAP fault.
VersionData	The details about a version of a file on the protocol server.

2.2.4.1 Results

The **Results** complex type provides the details about all versions of a specified file that the user can access.

The **DeleteAllVersions**, **DeleteVersion**, **GetVersions**, and **RestoreVersion** methods return the **Results** complex type.

```
<s:complexType name="Results">
  <s:sequence>
    <s:element name="list" maxOccurs="1" minOccurs="1">
      <s:complexType>
        <s:attribute name="id" type="s:string" use="required" />
      </s:complexType>
    </s:element>
    <s:element name="versioning" maxOccurs="1" minOccurs="1">
      <s:complexType>
        <s:attribute name="enabled" type="s:unsignedByte" use="required" />
      </s:complexType>
    </s:element>
    <s:element name="settings" maxOccurs="1" minOccurs="1">
      <s:complexType>
        <s:attribute name="url" type="s:string" use="required" />
      </s:complexType>
    </s:element>
    <s:element name="result" maxOccurs="unbounded" minOccurs="1" type="tns:VersionData"/>
  </s:sequence>
</s:complexType>
```

list: Contains information about the **document library** that is represented by this **list**.

list.id: Specifies the **GUID** of the document library in which the file resides.

versioning: Contains version-related information about the file.

versioning.enabled: Specifies whether versioning is enabled on the file. The value of this attribute MUST be "0" or "1". A value of "0" indicates that versioning is disabled; a value of "1" indicates that versioning is enabled.

settings: Contains information about how versioning operates for the file.

settings.url: Specifies the URL to the webpage of versioning-related settings for the document library in which the file resides.

result: Specifies details about a version of the file on the protocol server. A separate **result** element MUST exist for each version of the file that the user can access. For details about the type of this element, see section 2.2.4.3.

2.2.4.2 SOAPFaultDetails

The **SOAPFaultDetails** complex type specifies the details about a SOAP fault.

```
<s:schema xmlns:s="http://www.w3.org/2001/XMLSchema" targetNamespace="
http://schemas.microsoft.com/sharepoint/soap">
   <s:complexType name="SOAPFaultDetails">
      <s:sequence>
         <s:element name="errorstring" type="s:string"/>
         <s:element name="errorcode" type="s:string" minOccurs="0"/>
      </s:sequence>
   </s:complexType>
</s:schema>
```

errorstring: Human-readable text that explains the application-level fault.

errorcode: The hexadecimal representation of a 4-byte result code.

2.2.4.3 VersionData

The **VersionData** complex type specifies the details about a single version of a file.

```
<s:complexType name="VersionData">
  <s:attribute name="version" type="s:string" use="required" />
  <s:attribute name="url" type="s:string" use="required" />
  <s:attribute name="created" type="s:string" use="required" />
  <s:attribute name="createdRaw" type="s:string" use="required" />
  <s:attribute name="createdBy" type="s:string" use="required" />
  <s:attribute name="createdByName" type="s:string" use="optional" />
  <s:attribute name="size" type="s:unsignedLong" use="required" />
  <s:attribute name="comments" type="s:string" use="required" />
</s:complexType>
```

version: The version of the file. The most recent version of the file MUST be preceded with an at sign (@). All the other versions MUST exist without any prefix. For example, if a file has two versions—1.0 and 2.0, with 2.0 being the most recent version of the file—the value of this attribute for the most recent file is "@2.0", and the value for the previous version is "1.0". The format SHOULD contain the version of the file, including the major version and minor version numbers connected by period, for example, "1.0".

url: The complete URL of the version of the file.

created: The creation date and time for the version of the file that is displayed, using an implementation-specific format.

createdRaw: The creation date and time for the version of the file in **Datetime** format, as specified in [ISO-8601].<1>

createdBy: The creator of the version of the file.

createdByName: The display name of the creator of the version of the file.

size: The size, in bytes, of the version of the file.

comments: The comment entered when the version of the file was replaced on the protocol server during **check in**.

2.2.5 Simple Types

This specification does not define any common XML schema simple type definitions.

2.2.6 Attributes

This specification does not define any common XML schema attribute definitions.

2.2.7 Groups

This specification does not define any common XML schema group definitions.

2.2.8 Attribute Groups

This specification does not define any common XML schema attribute group definitions.

3   Protocol Details

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.

Except where specified, protocol clients SHOULD interpret HTTP status codes returned by the protocol server as specified in [RFC2616] section 10.

This protocol allows protocol servers to provide additional details for SOAP faults by including either a **detail** element as specified in [SOAP1.1] section 4.4 or a **Detail** element as specified in [SOAP1.2/1] section 5.4.5. These elements conform to the **XSD** of the **SOAPFaultDetails** complex type specified in section 2.2.4.2. Except where specified, these SOAP faults are not significant for interoperability, and protocol clients can interpret them in an implementation-specific manner.

This protocol allows protocol servers to perform implementation-specific authorization checks and notify protocol clients of authorization faults either by using HTTP status codes or by using SOAP faults as specified previously in this section.

3.1   VersionsSoap Server Details

The server side of this protocol contains the following operations:

- **DeleteAllVersions**

- **DeleteVersion**

- **GetVersions**

- **RestoreVersion**

Information returned from the **GetVersions** operation is used to call the **DeleteVersion** and **RestoreVersion** operations. These operations are described in section 3.1.4.

3.1.1   Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This specification does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this specification.

For more details, see [MS-WSSTS] section 2.1.2.7, [MS-WSSTS] section 2.1.2.11, and [MS-WSSTS] section 2.1.2.11.1.3.

3.1.2   Timers

None.

3.1.3   Initialization

None.

3.1.4   Message Processing Events and Sequencing Rules

The following table summarizes the list of operations as defined by this specification.

Operation	Description
DeleteAllVersions	Deletes all the previous versions of the specified file.
DeleteVersion	Deletes the specified version of the file.
GetVersions	Gets details about all the versions of the specified file.
RestoreVersion	Restores the specified file to a specific version.

3.1.4.1 DeleteAllVersions

The **DeleteAllVersions** operation deletes all the previous versions of the specified file, except for the **published version** and the **current version**. If the **Recycle Bin** is enabled, the versions are placed in the Recycle Bin, instead.

```
<wsdl:operation name="DeleteAllVersions">
    <wsdl:input message="tns:DeleteAllVersionsSoapIn" />
    <wsdl:output message="tns:DeleteAllVersionsSoapOut" />
</wsdl:operation>
```

The protocol client sends a **DeleteAllVersionsSoapIn** request message, and the protocol server responds with a **DeleteAllVersionsSoapOut** response message.

3.1.4.1.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
DeleteAllVersionsSoapIn	The WSDL request message for a WSDL **DeleteAllVersions** operation.
DeleteAllVersionsSoapOut	The WSDL response message for a WSDL **DeleteAllVersions** operation.

3.1.4.1.1.1   DeleteAllVersionsSoapIn

The **DeleteAllVersionsSoapIn** message is the request message for deleting all previous versions of the specified file on the protocol server.

The **SOAP action** value of the message is:

http://schemas.microsoft.com/sharepoint/soap/DeleteAllVersions

The **SOAP body** contains a **DeleteAllVersions** element.

3.1.4.1.1.2   DeleteAllVersionsSoapOut

The **DeleteAllVersionsSoapOut** message is the response message for deleting all previous versions of the specified file on the protocol server.

The SOAP body contains a **DeleteAllVersionsResponse** element.

3.1.4.1.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
DeleteAllVersions	The input data for the **DeleteAllVersions WSDL operation**.
DeleteAllVersionsResponse	The result data for the **DeleteAllVersions** WSDL operation.

3.1.4.1.2.1   DeleteAllVersions

The **DeleteAllVersions** element specifies the input data for the **DeleteAllVersions** WSDL operation.

```
<s:element name="DeleteAllVersions">
  <s:complexType>
    <s:sequence>
      <s:element name="fileName" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

fileName: An element as specified in section 2.2.3.1.

3.1.4.1.2.2   DeleteAllVersionsResponse

The **DeleteAllVersionsResponse** element specifies the result data for the **DeleteAllVersions** WSDL operation.

```
<s:element name="DeleteAllVersionsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="DeleteAllVersionsResult">
        <s:complexType>
          <s:sequence>
            <s:element name="results" minOccurs="1" maxOccurs="1" type="tns:Results" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

DeleteAllVersionsResult: An **XML node** that conforms to the structure specified in section 2.2.4.1 and that contains the details about all the versions of the specified file that the user can access.

The error conditions that can be encountered during this operation generate the error codes that are shown in the following table.

Error code	Description
No error code element returned	The system cannot find the specified file. Exception from HRESULT: 0x80070002.<2>
0x81020030	This error code SHOULD<3> indicate that the **fileName** element of the **DeleteAllVersions** element contains invalid characters.<4>

Note that invalid characters are defined as the following:

- Double quotation mark (")

- Pound sign (#)

- Percent sign (%)

- Ampersand (&)

- Asterisk (*)

- Colon (:)

- Left angle bracket (<)

- Equal sign (=)

- Right angle bracket (>)

- Question mark (?)

- Backward slash (\)

- Left curly bracket ({)

- Vertical bar (|)

- Right curly bracket (})

- Tab (\t)

- Tilde (~)

- Two consecutive dots (..)

- Any character that has a value less than 32 or greater than 126

3.1.4.1.3 Complex Types

None.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 DeleteVersion

The **DeleteVersion** operation deletes a specific version of the specified file. If the Recycle Bin is enabled, the version is placed in the Recycle Bin, instead. This operation MUST NOT grant deletion of the current version or the published version of the file.

```
<wsdl:operation name="DeleteVersion">
    <wsdl:input message="tns:DeleteVersionSoapIn" />
    <wsdl:output message="tns:DeleteVersionSoapOut" />
</wsdl:operation>
```

The protocol client sends a **DeleteVersionSoapIn** request message, and the protocol server responds with a **DeleteVersionSoapOut** response message.

3.1.4.2.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
DeleteVersionSoapIn	The WSDL request message for a WSDL **DeleteVersion** operation.
DeleteVersionSoapOut	The WSDL response message for a WSDL **DeleteVersion** operation.

3.1.4.2.1.1   DeleteVersionSoapIn

The **DeleteVersionSoapIn** message is the request message for deleting a specific version of the specified file on the protocol server.

The SOAP action value of the message is:

http://schemas.microsoft.com/sharepoint/soap/DeleteVersion

The SOAP body contains a **DeleteVersion** element.

3.1.4.2.1.2   DeleteVersionSoapOut

The **DeleteVersionSoapOut** message is the response message for deleting a version of the specified file on the protocol server.

The SOAP body contains a **DeleteVersionResponse** element.

3.1.4.2.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description

Element	Description
DeleteVersion	The input data for the **DeleteVersion** WSDL operation.
DeleteVersionResponse	The result data for the **DeleteVersion** WSDL operation.

3.1.4.2.2.1   DeleteVersion

The **DeleteVersion** element specifies the input data for the **DeleteVersion** WSDL operation.

```
<s:element name="DeleteVersion">
  <s:complexType>
    <s:sequence>
      <s:element name="fileName" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="fileVersion" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

fileName: An element as specified in section 2.2.3.1.

fileVersion: An element as specified in section 2.2.3.2.

3.1.4.2.2.2   DeleteVersionResponse

The **DeleteVersionResponse** element specifies the result data for the **DeleteVersion** WSDL operation.

```
<s:element name="DeleteVersionResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="DeleteVersionResult">
        <s:complexType>
          <s:sequence>
            <s:element name="results" minOccurs="1" maxOccurs="1" type="tns:Results" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

DeleteVersionResult: An XML node that conforms to the structure specified in section 2.2.4.1 and that contains the details about all the versions of the specified file that the user can access.

The error conditions that can be encountered during this operation generate the error codes that are shown in the following table.

Error code	Description
0x81020073	This error code SHOULD<5> indicate that the **fileName** element of the **DeleteVersion** element contains invalid characters.<6>
0x80131600	The specified version does not exist or was recently deleted. This error code is returned when either the value specified by the **fileVersion** element is the current or published version of the file or the file does not exist.<7>

Note that invalid characters are defined as the following:

- Double quotation mark (")

- Pound sign (#)

- Percent sign (%)

- Ampersand (&)

- Asterisk (*)

- Colon (:)

- Left angle bracket (<)

- Equal sign (=)

- Right angle bracket (>)

- Question mark (?)

- Backward slash (\)

- Left curly bracket ({)

- Vertical bar (|)

- Right curly bracket (})

- Tilde (~)

- Forward slash (/)

- Two consecutive dots (..)

- Any character that has a value less than 32 or greater than 126

3.1.4.2.3 Complex Types

None.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.4.3 GetVersions

The **GetVersions** operation gets details about all versions of the specified file that the user can access.

```
<wsdl:operation name="GetVersions">
    <wsdl:input message="tns:GetVersionsSoapIn" />
    <wsdl:output message="tns:GetVersionsSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetVersionsSoapIn** request message, and the protocol server responds with a **GetVersionsSoapOut** response message.

3.1.4.3.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetVersionsSoapIn	The WSDL request message for a **GetVersions** WSDL operation.
GetVersionsSoapOut	The WSDL response message for a **GetVersions** WSDL operation.

3.1.4.3.1.1   GetVersionsSoapIn

The **GetVersionsSoapIn** message is the request message for getting all versions of the specified file that the user can access.

The SOAP action value of the message is:

http://schemas.microsoft.com/sharepoint/soap/GetVersions

The SOAP body contains a **GetVersions** element.

3.1.4.3.1.2   GetVersionsSoapOut

The **GetVersionsSoapOut** message is the response message for getting all versions of the specified file that the user can access.

The SOAP body contains a **GetVersionsResponse** element.

3.1.4.3.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation

Element	Description
GetVersions	The input data for the **GetVersions** WSDL operation.
GetVersionsResponse	The result data for the **GetVersions** WSDL operation.

3.1.4.3.2.1 GetVersions

The **GetVersions** element specifies the input data for the **GetVersions** WSDL operation.

```
<s:element name="GetVersions">
  <s:complexType>
    <s:sequence>
      <s:element name="fileName" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

fileName: An element as specified in section 2.2.3.1.

3.1.4.3.2.2 GetVersionsResponse

The **GetVersionsResponse** element specifies the result data for the **GetVersions** WSDL operation.

```
<s:element name="GetVersionsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetVersionsResult">
        <s:complexType>
          <s:sequence>
            <s:element name="results" minOccurs="1" maxOccurs="1" type="tns:Results" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetVersionsResult: An XML node that conforms to the structure specified in section 2.2.4.1 and that contains the details about all the versions of the specified file that the user can access.

The error conditions that can be encountered during this operation generate the error code that is shown in the following table.

Error code	Description
No error code element returned	The system cannot find the specified file. Exception from HRESULT: 0x80070002.

3.1.4.3.3 Complex Types

None.

3.1.4.3.4 Simple Types

None.

3.1.4.3.5 Attributes

None.

3.1.4.3.6 Groups

None.

3.1.4.3.7 Attribute Groups

None.

3.1.4.4 RestoreVersion

The **RestoreVersion** operation restores the specified file to a specific version. After the restoration, the current version number of the file MUST still be increased, as with any other change.

```
<wsdl:operation name="RestoreVersion">
    <wsdl:input message="tns:RestoreVersionSoapIn" />
    <wsdl:output message="tns:RestoreVersionSoapOut" />
</wsdl:operation>
```

When restoring a version, the version MUST NOT be in the Recycle Bin and MUST NOT have been deleted.

If the file is not already at the **checked out publishing level**, and the protocol server enforces that only checked out files can be modified, this operation MUST NOT allow the restoration of the file.

If the protocol server enforces that only checked out files can be modified, the protocol client calls the **CheckOutFile** operation, as specified in [MS-LISTSWS] section 3.1.4.8, before calling this operation.

After calling this operation, the protocol client calls either the **CheckInFile** operation, as specified in [MS-LISTSWS] section 3.1.4.7, to commit the changes performed by this operation, or the **UndoCheckOut** operation, as specified in [MS-LISTSWS] section 3.1.4.26, to revert the changes performed by this operation.

If check out is not being enforced, it is not necessary to check out the file before the restoration, and it is not necessary to check in the file after the restoration. If the file is not checked out before the restoration, the current version number of the file MUST still be increased, as with any other change. If the file is checked out, the current version number of the file after restoration MUST remain the same as before restoration.

The protocol client sends a **RestoreVersionSoapIn** request message, and the protocol server responds with a **RestoreVersionSoapOut** response message.

3.1.4.4.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
RestoreVersionSoapIn	The WSDL request message for a WSDL **RestoreVersion** operation.
RestoreVersionSoapOut	The WSDL response message for a WSDL **RestoreVersion** operation.

3.1.4.4.1.1   RestoreVersionSoapIn

The **RestoreVersionSoapIn** message is the request message for restoring the specified file to a specific version.

The SOAP action value of the message is:

http://schemas.microsoft.com/sharepoint/soap/RestoreVersion

The SOAP body contains a **RestoreVersion** element.

3.1.4.4.1.2   RestoreVersionSoapOut

The **RestoreVersionSoapOut** message is the response message for restoring the specified file to a specific version.

The SOAP body contains a **RestoreVersionResponse** element.

3.1.4.4.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation

Element	Description
RestoreVersion	The input data for the **RestoreVersion** WSDL operation.
RestoreVersionResponse	The result data for the **RestoreVersion** WSDL operation.

3.1.4.4.2.1   RestoreVersion

The **RestoreVersion** element specifies the input data for the **RestoreVersion** WSDL operation.

```
<s:element name="RestoreVersion">
  <s:complexType>
    <s:sequence>
      <s:element name="fileName" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="fileVersion" type="s:string" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

fileName: An element as specified in section 2.2.3.1.

fileVersion: An element as specified in section 2.2.3.2.

3.1.4.4.2.2   RestoreVersionResponse

The **RestoreVersionResponse** element specifies the result data for the **RestoreVersion** WSDL operation.

```
<s:element name="RestoreVersionResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="RestoreVersionResult">
        <s:complexType>
          <s:sequence>
            <s:element name="results" minOccurs="1" maxOccurs="1" type="tns:Results" />
          </s:sequence>
```

```
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
```

RestoreVersionResult: MUST return an XML node that conforms to the structure specified in section 2.2.4.1, which contains the details about all the versions of the specified file that the user can access.

The error conditions that can be encountered during this operation generate the error codes that are shown in the following table.

Error code	Description
No error code element returned	The file specified by the **fileName** and **fileVersion** elements of the **RestoreVersion** element could not be found on the site.<8>
0x8007009E	The file is not checked out.<9> This file needs to be checked out before changes can be made.
0x81020073	This error code SHOULD<10> indicate that the **fileName** file or folder name contains invalid characters.<11>

Note that invalid characters are defined as the following:

▪ Percent sign (%)

▪ Ampersand (&)

▪ Asterisk (*)

▪ Colon (:)

▪ Left angle bracket (<)

▪ Right angle bracket (>)

▪ Question mark (?)

▪ Forward slash (/)

▪ Left curly bracket ({)

▪ Vertical bar (|)

▪ Right curly bracket (})

▪ Tilde (~)

▪ Double quotation mark (")

▪ Backward slash (\)

▪ Two consecutive dots (..)

▪ Any character that has a value less than 32 or greater than 126

3.1.4.4.3 Complex Types

None.

3.1.4.4.4 Simple Types

None.

3.1.4.4.5 Attributes

None.

3.1.4.4.6 Groups

None.

3.1.4.4.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4   Protocol Examples

This example demonstrates how to restore a file to its first version.

In this example, a site named TestSite contains a folder hierarchy of the form RootFolder/SubFolder on the protocol server. Three versions of a file named Test.txt exist in the folder named SubFolder. The versions are 0.1, 0.2, and 0.3, where 0.1 is the first version that was created, and 0.3 is the current version. The protocol client needs to restore the file to version 0.1. The following steps illustrate the process.

1. To obtain the current list of versions for the Test.txt file, the protocol client issues a **GetVersions** request to TestSite, specifying the file location relative to the site (that is, RootFolder/SubFolder/Test.txt), as follows:

```
<GetVersions xmlns="http://schemas.microsoft.com/sharepoint/soap/">
    <fileName>RootFolder/SubFolder/Test.txt</fileName>
</GetVersions>
```

2. The protocol server returns a list of the file versions in response. The response contains an XML node that is a **Results** complex type:

```
<GetVersionsResponse xmlns="http://schemas.microsoft.com/sharepoint/soap/">
<GetVersionsResult>
    <results xmlns="http://schemas.microsoft.com/sharepoint/soap/">
    <list id="{1D3EBE41-D0D9-4431-BE41-44311D3ED0D9}" />
     <versioning enabled="1" />
     <settings url="http://server/TestSite/_layouts/LstSetng.aspx?List={1D3EBE41-D0D9-4431-
BE41-44311D3ED0D9}" />
     <result version="@0.3" url="http://server/TestSite/RootFolder/SubFolder/Test.txt"
created="3/10/2010 4:00 PM" createdRaw="2010-03-11T00:00:31Z" createdBy="DOMAIN\user"
createdByName="User" size="8" comments="upload 3" />
     <result version="0.1"
url="http://server/TestSite/_vti_history/1/RootFolder/SubFolder/Test.txt" created="3/10/2010
3:58 PM" createdRaw="2010-03-10T23:58:55Z" createdBy="DOMAIN\user" createdByName="User"
size="7" comments="upload 1" />
            <result version = "0.2"
url="http://server/TestSite/ vti history/2/RootFolder/SubFolder/Test.txt" created="3/10/2010
4:00 PM" createdRaw="2010-03-11T00:00:03Z" createdBy="DOMAIN\user" createdByName="User"
size="8" comments="upload 2" />
    </results>
</GetVersionsResult>
</GetVersionsResponse>
```

3. The protocol client parses the response and finds the first version of the file, which is 0.1.

4. The protocol client issues a **CheckOutFile** request, as described in [MS-LISTSWS] section 3.1.4.8, specifying the file name as "RootFolder/SubFolder/Test.txt".

```
<CheckOutFile xmlns="http://schemas.microsoft.com/sharepoint/soap/">
  <pageUrl>http://server/TestSite/RootFolder/SubFolder/Test.txt</pageUrl>
  <checkoutToLocal>true</checkoutToLocal>
  <lastmodified></lastmodified>
</CheckOutFile>
```

5. The protocol client issues a **RestoreVersion** request, specifying the file name as "RootFolder/SubFolder/Test.txt" and the version as "0.1".

```
<RestoreVersion xmlns="http://schemas.microsoft.com/sharepoint/soap/">
  <fileName>RootFolder/SubFolder/Test.txt</fileName>
  <fileVersion>0.1</fileVersion>
</RestoreVersion>
```

6. The protocol server restores the current version of the file to the specified version.

7. The protocol client issues a **CheckInFile** request, as described in [MS-LISTSWS] section 3.1.4.7, for the file, with the **CheckinType** element, as described in [MS-LISTSWS] section 3.1.4.7.2.1, set to "MinorCheckin".

```
<CheckInFile xmlns="http://schemas.microsoft.com/sharepoint/soap/">
  <pageUrl>http://server/TestSite/RootFolder/SubFolder/Test.txt</pageUrl>
  <comment>Restore</comment>
  <CheckinType>MinorCheckin</CheckinType>
</CheckInFile>
```

8. The protocol server now has four versions of the file, namely "0.1", "0.2", "0.3", and "0.4". The contents of the files that have the versions "0.4" and "0.1" are identical.

5   Security

5.1   Security Considerations for Implementers

None.

5.2   Index of Security Parameters

None.

6   Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided in this appendix.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://schemas.microsoft.com/sharepoint/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/">
      <s:import namespace="http://www.w3.org/2001/XMLSchema" />
      <s:complexType name="VersionData">
        <s:attribute name="version" type="s:string" use="required" />
        <s:attribute name="url" type="s:string" use="required" />
        <s:attribute name="created" type="s:string" use="required" />
        <s:attribute name="createdRaw" type="s:string" use="required" />
        <s:attribute name="createdBy" type="s:string" use="required" />
        <s:attribute name="createdByName" type="s:string" use="optional" />
        <s:attribute name="size" type="s:unsignedLong" use="required" />
        <s:attribute name="comments" type="s:string" use="required" />
      </s:complexType>
      <s:complexType name="Results">
        <s:sequence>
          <s:element name="list" maxOccurs="1" minOccurs="1">
            <s:complexType>
              <s:attribute name="id" type="s:string" use="required" />
            </s:complexType>
          </s:element>
          <s:element name="versioning" maxOccurs="1" minOccurs="1">
            <s:complexType>
              <s:attribute name="enabled" type="s:unsignedByte" use="required" />
            </s:complexType>
          </s:element>
          <s:element name="settings" maxOccurs="1" minOccurs="1">
            <s:complexType>
              <s:attribute name="url" type="s:string" use="required" />
            </s:complexType>
          </s:element>
          <s:element maxOccurs="unbounded" minOccurs="1" name="result"
type="tns:VersionData"/>
        </s:sequence>
      </s:complexType>
      <s:element name="GetVersions">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="fileName" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetVersionsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetVersionsResult">
              <s:complexType>
                <s:sequence>
                  <s:element name="results" minOccurs="1" maxOccurs="1" type="tns:Results" />
                </s:sequence>
              </s:complexType>
```

```
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="RestoreVersion">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="fileName" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="fileVersion" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="RestoreVersionResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="RestoreVersionResult">
            <s:complexType>
              <s:sequence>
                <s:element name="results" minOccurs="1" maxOccurs="1" type="tns:Results" />
              </s:sequence>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="DeleteVersion">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="fileName" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="fileVersion" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="DeleteVersionResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="DeleteVersionResult">
            <s:complexType>
              <s:sequence>
                <s:element name="results" minOccurs="1" maxOccurs="1" type="tns:Results" />
              </s:sequence>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="DeleteAllVersions">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="fileName" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="DeleteAllVersionsResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="DeleteAllVersionsResult">
            <s:complexType>
              <s:sequence>
                <s:element name="results" minOccurs="1" maxOccurs="1" type="tns:Results" />
              </s:sequence>
            </s:complexType>
          </s:element>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:schema>
</wsdl:types>
<wsdl:message name="GetVersionsSoapIn">
```

```
      <wsdl:part name="parameters" element="tns:GetVersions" />
    </wsdl:message>
    <wsdl:message name="GetVersionsSoapOut">
      <wsdl:part name="parameters" element="tns:GetVersionsResponse" />
    </wsdl:message>
    <wsdl:message name="RestoreVersionSoapIn">
      <wsdl:part name="parameters" element="tns:RestoreVersion" />
    </wsdl:message>
    <wsdl:message name="RestoreVersionSoapOut">
      <wsdl:part name="parameters" element="tns:RestoreVersionResponse" />
    </wsdl:message>
    <wsdl:message name="DeleteVersionSoapIn">
      <wsdl:part name="parameters" element="tns:DeleteVersion" />
    </wsdl:message>
    <wsdl:message name="DeleteVersionSoapOut">
      <wsdl:part name="parameters" element="tns:DeleteVersionResponse" />
    </wsdl:message>
    <wsdl:message name="DeleteAllVersionsSoapIn">
      <wsdl:part name="parameters" element="tns:DeleteAllVersions" />
    </wsdl:message>
    <wsdl:message name="DeleteAllVersionsSoapOut">
      <wsdl:part name="parameters" element="tns:DeleteAllVersionsResponse" />
    </wsdl:message>
    <wsdl:portType name="VersionsSoap">
      <wsdl:operation name="GetVersions">
        <wsdl:input message="tns:GetVersionsSoapIn" />
        <wsdl:output message="tns:GetVersionsSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="RestoreVersion">
        <wsdl:input message="tns:RestoreVersionSoapIn" />
        <wsdl:output message="tns:RestoreVersionSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="DeleteVersion">
        <wsdl:input message="tns:DeleteVersionSoapIn" />
        <wsdl:output message="tns:DeleteVersionSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="DeleteAllVersions">
        <wsdl:input message="tns:DeleteAllVersionsSoapIn" />
        <wsdl:output message="tns:DeleteAllVersionsSoapOut" />
      </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="VersionsSoap" type="tns:VersionsSoap">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
      <wsdl:operation name="GetVersions">
        <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetVersions"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="RestoreVersion">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/RestoreVersion" style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="DeleteVersion">
        <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteVersion"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
```

```
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="DeleteAllVersions">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteAllVersions" style="document"
/>
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="VersionsSoap12" type="tns:VersionsSoap">
      <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
      <wsdl:operation name="GetVersions">
        <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/GetVersions"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="RestoreVersion">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/RestoreVersion" style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="DeleteVersion">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteVersion" style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="DeleteAllVersions">
        <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/DeleteAllVersions" style="document"
/>
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
</wsdl:definitions>
```

7   Appendix B: Product Behavior

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

- The 2007 Microsoft Office system

- Microsoft Office 2010 suites

- Microsoft Office 2013

- Microsoft SharePoint Foundation 2010

- Microsoft SharePoint Foundation 2013

- Windows SharePoint Services 2.0

- Windows SharePoint Services 3.0

- Microsoft Office 2016

- Microsoft SharePoint Server 2016

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

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

<1> Section 2.2.4.3: Windows SharePoint Services 3.0 does not return this attribute.

<2> Section 3.1.4.1.2.2: Windows SharePoint Services 3.0 returns the 0x81070906 error code, and SharePoint Foundation 2010 returns no error code.

<3> Section 3.1.4.1.2.2: Windows SharePoint Services 3.0 returns the 0x81070905 error code for invalid characters.

<4> Section 3.1.4.1.2.2: Windows SharePoint Services truncates all characters after the first pound sign (#) or question mark (?) in the file name. If the truncated file name does not exist on the server, SharePoint Foundation 2010 and SharePoint Foundation 2013 returns a SOAP server exception, and Windows SharePoint Services 3.0 returns the 0x81070906 error code. Tab (\t) is ignored at the end of the **fileName**. The backward slash (\) is ignored at the end of the **fileName** on Windows SharePoint Services 3.0, SharePoint Foundation 2010, and SharePoint Foundation 2013. Windows SharePoint Services 3.0, SharePoint Foundation 2010, and SharePoint Foundation 2013 return different error codes depending on where the backward slash (\) occurs in the **fileName**. If **fileName** contains equals sign (=), Windows SharePoint Services 3.0 returns error code 0x81070906, and SharePoint Foundation 2010 and SharePoint Foundation 2013 return a SOAP exception without an error code.

<5> Section 3.1.4.2.2.2: Windows SharePoint Services 3.0 returns the 0x81070970 error code for invalid characters.

<6> Section 3.1.4.2.2.2: Windows SharePoint Services truncates all characters after the first pound sign (#) or question mark (?) in the file name. If the truncated file name does not exist on the server, 0x80131600 is returned, except that Windows SharePoint Services 3.0 returns a SOAP exception. The tab (\t) character is ignored at the end of the **fileName** on Windows SharePoint Services 3.0, SharePoint Foundation 2010,and SharePoint Foundation 2013. The backward slash (\) character is

ignored at the end of the **fileName** on Windows SharePoint Services 3.0, SharePoint Foundation 2010, and SharePoint Foundation 2013. Windows SharePoint Services 3.0, SharePoint Foundation 2010, and SharePoint Foundation 2013 return different error codes depending on where the backward slash (\) occurs in the file name. If **fileName** contains an equal sign (=), Windows SharePoint Services 3.0 returns a SOAP exception without an error code; SharePoint Foundation 2010 and SharePoint Foundation 2013 return error code 0x80131600.

<7> Section 3.1.4.2.2.2: Windows SharePoint Services 3.0 does not return an error code element but does return an error string of "Object reference not set to an instance of an object."

<8> Section 3.1.4.4.2.2: Windows SharePoint Services 3.0 returns error code 0x81070906 with the description, "There is no file with URL '**fileName**' in this Web."

<9> Section 3.1.4.4.2.2: Windows SharePoint Services 3.0 returns error code 0x81070975.

<10> Section 3.1.4.4.2.2: Windows SharePoint Services 3.0, SharePoint Foundation 2010, and SharePoint Foundation 2013 return different error codes depending on where the pound sign (#) or question mark (?) character occurs in the **fileName**. In Windows SharePoint Services 3.0, SharePoint Foundation 2010 and SharePoint Foundation 2013 two consecutive dots (..) are ignored at the end of the **fileName** or folder name. In Windows SharePoint Services 3.0, SharePoint Foundation 2010, and SharePoint Foundation 2013, the forward slash (/) character is ignored at the end of the **fileName**. Windows SharePoint Services 3.0, SharePoint Foundation 2010, and SharePoint Foundation 2013 return different error codes depending on where the forward slash (/) occurs in the file specification.

<11> Section 3.1.4.4.2.2: Windows SharePoint Services 3.0 returns error code 0x81070970 for invalid characters with the description, "The file or folder name "fileName" contains invalid characters. Please use a different name. Common invalid characters include the following: # % & *: < > ? / { | }".

8 Change Tracking

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

9 Index