

[MS-VERSS]: Versions Web Service Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
10/06/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
07/13/2009	1.03	Major	Revised and edited the technical content
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	Editorial	Revised and edited the technical content
02/19/2010	2.0	Editorial	Revised and edited the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Minor	Clarified the meaning of the technical content.
07/23/2010	2.05	Editorial	Changed language and formatting in the technical content.
09/27/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	5
1.2.1 Normative References	5
1.2.2 Informative References	6
1.3 Protocol Overview (Synopsis)	6
1.4 Relationship to Other Protocols	7
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	8
1.8 Vendor-Extensible Fields	8
1.9 Standards Assignments	8
2 Messages	9
2.1 Transport	9
2.2 Common Message Syntax	9
2.2.1 Namespaces	9
2.2.2 Messages	9
2.2.3 Elements	10
2.2.3.1 fileName	10
2.2.3.2 fileVersion	10
2.2.4 Complex Types	10
2.2.4.1 Results	11
2.2.4.2 SOAPFaultDetails	11
2.2.4.3 VersionData	12
2.2.5 Simple Types	12
2.2.6 Attributes	13
2.2.7 Groups	13
2.2.8 Attribute Groups	13
3 Protocol Details	14
3.1 VersionsSoap Server Details	14
3.1.1 Abstract Data Model	14
3.1.2 Timers	14
3.1.3 Initialization	14
3.1.4 Message Processing Events and Sequencing Rules	14
3.1.4.1 DeleteAllVersions	15
3.1.4.1.1 Messages	15
3.1.4.1.1.1 DeleteAllVersionsSoapIn	15
3.1.4.1.1.2 DeleteAllVersionsSoapOut	15
3.1.4.1.2 Elements	16
3.1.4.1.2.1 DeleteAllVersions	16
3.1.4.1.2.2 DeleteAllVersionsResponse	16
3.1.4.1.3 Complex Types	17
3.1.4.1.4 Simple Types	17
3.1.4.1.5 Attributes	17
3.1.4.1.6 Groups	17
3.1.4.1.7 Attribute Groups	17
3.1.4.2 DeleteVersion	17
3.1.4.2.1 Messages	17

3.1.4.2.1.1	DeleteVersionSoapIn	17
3.1.4.2.1.2	DeleteVersionSoapOut	18
3.1.4.2.2	Elements	18
3.1.4.2.2.1	DeleteVersion	18
3.1.4.2.2.2	DeleteVersionResponse	18
3.1.4.2.3	Complex Types	19
3.1.4.2.4	Simple Types	19
3.1.4.2.5	Attributes	19
3.1.4.2.6	Groups	19
3.1.4.2.7	Attribute Groups	19
3.1.4.3	GetVersions	19
3.1.4.3.1	Messages	20
3.1.4.3.1.1	GetVersionsSoapIn	20
3.1.4.3.1.2	GetVersionsSoapOut	20
3.1.4.3.2	Elements	20
3.1.4.3.2.1	GetVersions	20
3.1.4.3.2.2	GetVersionsResponse	21
3.1.4.3.3	Complex Types	21
3.1.4.3.4	Simple Types	21
3.1.4.3.5	Attributes	21
3.1.4.3.6	Groups	21
3.1.4.3.7	Attribute Groups	21
3.1.4.4	RestoreVersion	21
3.1.4.4.1	Messages	22
3.1.4.4.1.1	RestoreVersionSoapIn	22
3.1.4.4.1.2	RestoreVersionSoapOut	22
3.1.4.4.2	Elements	23
3.1.4.4.2.1	RestoreVersion	23
3.1.4.4.2.2	RestoreVersionResponse	23
3.1.4.4.3	Complex Types	24
3.1.4.4.4	Simple Types	24
3.1.4.4.5	Attributes	24
3.1.4.4.6	Groups	24
3.1.4.4.7	Attribute Groups	24
3.1.5	Timer Events	24
3.1.6	Other Local Events	24
4	Protocol Examples	25
4.1	Restoring a File to Its First Version	25
5	Security	27
5.1	Security Considerations for Implementers	27
5.2	Index of Security Parameters	27
6	Appendix A: Full WSDL	28
7	Appendix B: Product Behavior	33
8	Change Tracking	34
9	Index	35

1 Introduction

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

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

globally unique identifier (GUID)
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)

The following terms are defined in [\[MS-OFCGLOS\]](#):

check in
checked out
current version
document library
file
folder
list
published version
publishing level
Recycle Bin
site
SOAP (Simple Object Access Protocol)
SOAP action
SOAP body
SOAP fault
SOAP operation
URL (Uniform Resource Locator)
WSDL (Web Services Description Language)
WSDL operation
XML attribute
XML element
XML namespace
XML namespace prefix
XML node
XML Schema

The following terms are specific to this document:

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-LISTSWS] Microsoft Corporation, "[Lists Web Service Protocol Specification](#)", June 2008.

[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., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.ietf.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] World Wide Web Consortium, "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation 8 December 2009, <http://www.w3.org/TR/REC-xml-names/>

[XMLSCHEMA1] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "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

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

[MS-OFCGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)", June 2008.

[MS-WSSTS] Microsoft Corporation, "[Windows SharePoint Services Technical Specification](#)", June 2008.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <http://www.ietf.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 retrieval of all versions of a file, deletion of previous versions of a file, and restoration of a file to a previous version.

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: Versions SOAP 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 **HTTPS**, as described in [\[RFC2818\]](#).

The following diagram shows the underlying messaging and transport stack used by the protocol.

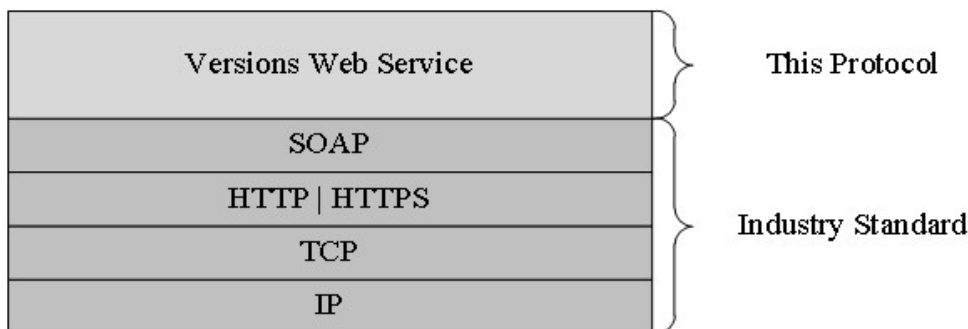


Figure 2: This protocol in relation to other protocols

This protocol calls the **CheckOutFile**, **CheckInFile**, and **UndoCheckOut** operations, as specified in [\[MS-LISTSWS\]](#).

1.5 Prerequisites/Preconditions

This protocol operates against a **site (2)** 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 (2). The following example is 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 document covers versioning issues in one area. 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 either in [\[SOAP1.1\]](#) section 4 (SOAP Envelope) or in [\[SOAP1.2/1\]](#) section 5 (SOAP Message Construct). Protocol server faults **MUST** be returned either via HTTP status codes as specified in [\[RFC2616\]](#) section 10 (Status Code Definitions), or via **SOAP faults** as specified either in [\[SOAP1.1\]](#) section 4.4 (SOAP Fault) or in [\[SOAP1.2/1\]](#) section 5.4 (SOAP Fault).

2.2 Common Message Syntax

This section contains common structures that this protocol uses. The syntax of the structures uses **XML Schema (1)**, as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and Web Services Description Language, as specified in [\[WSDL\]](#).

2.2.1 Namespaces

This specification defines and references **XML namespaces** by using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates an **XML namespace prefix** for each XML namespace that is used (as shown in the following table), 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/	
Wsd	http://schemas.xmlsoap.org/wsdl/	[WSDL]
(none)	http://schemas.microsoft.com/sharepoint/soap/	

2.2.2 Messages

None.

2.2.3 Elements

The following table summarizes the set of common XML Schema (2) element definitions defined in this specification. XML Schema (2) 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. The **fileName** element MUST be specified in one of the following formats:

- URL of the file
- Location of the file relative to the site (2) that contains it

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

- *folder1* and *folder2* are names of **folders**; *folder1* is the topmost folder in the site.
- *file-name* is the name of the file.

The following example shows this element.

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

2.2.3.2 fileVersion

The **fileVersion** element defines the version number of a file on the protocol server.

The following example shows this element.

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

2.2.4 Complex Types

The following table summarizes the set of common XML Schema (2) complex type definitions defined by this specification. XML Schema (2) complex type definitions that are specific to a particular operation are described with the operation.

Complex type	Description
Results	Specifies details about all versions of a file on the protocol server.
SOAPFaultDetails	Specifies the details of a SOAP fault.
VersionData	Specifies details about a version of a file on the protocol server.

2.2.4.1 Results

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

The methods [DeleteAllVersions](#), [DeleteVersion](#), [GetVersions](#), and [RestoreVersion](#) return the **Results** complex type. The following XML Schema (2) defines the basic structure of this 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; "1" indicates that versioning is enabled.

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

settings.url: Specifies the URL to the Web page of versioning-related settings of 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 the result element, see VersionData Complex Type (section [2.2.4.3](#)).

2.2.4.2 SOAPFaultDetails

The **SOAPFaultDetails** complex type specifies the details of a SOAP fault. The following XML Schema (2) defines the basic structure of this complex type.

```
<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 contains a set of **XML attributes**. It is used exclusively to specify details about a single version of a file. The following XML Schema (2) defines the basic structure of this complex type.

```

<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: Specifies the version of the file. The most recent version of the file **MUST** be preceded with an at sign (@). All the other versions **MUST** be 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 older version is "1.0".

url: Specifies 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: Specifies the creation date and time for the version of the file in ISO8601 Datetime format.

createdBy: Specifies the creator of the version of the file.

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

size: Specifies the size of the version of the file in bytes.

comments: Specifies the comment mentioned 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 (2) simple type definitions.

2.2.6 Attributes

This specification does not define any common XML Schema (2) attribute definitions.

2.2.7 Groups

This specification does not define any common XML Schema (2) group definitions.

2.2.8 Attribute Groups

This specification does not define any common XML Schema (2) 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 that the higher-layer protocol or application make are passed directly to the transport, and the results that the transport returns 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. This element conforms to the XML Schema (2) 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

Please reference [\[MS-WSSTS\]](#) section 2.1.2.7, [\[MS-WSSTS\]](#) section 2.1.2.11, and [\[MS-WSSTS\]](#) section 2.1.2.11.1.3 for details.

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 **WSDL operations** 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 is used to delete all the previous versions of the specified file except the **published version** and the **current version**. If the **Recycle Bin** is enabled, the versions are placed in the Recycle Bin instead. The following example shows this operation.

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

If versioning is disabled, the DeleteAllVersions operation MUST return information about the file without deleting any versions.

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 **WSDL** message definitions are specific to this 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 defined as follows.

```
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 action value of the message is defined as follows.

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

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

3.1.4.1.2 Elements

The following XML Schema (2) element definitions are specific to this operation.

3.1.4.1.2.1 DeleteAllVersions

The definition of the **DeleteAllVersions** element is as follows.

```
<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: For information about this element, see section [2.2.3.1](#).

3.1.4.1.2.2 DeleteAllVersionsResponse

The definition of the **DeleteAllVersionsResponse** element is as follows.

```
<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: Must return an **XML node** that conforms to the structure specified in section [2.2.4.1](#), that contains the details about all the versions of the specified file that the user can access.

Possible specific error conditions encountered during this operation generate the error codes shown in the following table.

Error code	Specification
	There is no file with URL filename in this Web .
0x81070906	The file specified by the fileName sub element in the DeleteAllVersions element could not be found in the site (2).
0x81020030	The fileName sub element in the DeleteAllVersions element contains invalid characters. Invalid characters defined in code are " # % & * : < = > ? \ { } ~ two consecutive dots and any character whose value is smaller than 32 or bigger than 126.

Error code	Specification
0x81070905	The fileName sub element in the DeleteAllVersions element contains invalid characters. Invalid characters defined in code are " # % & * : < = > ? \ { } ~ two consecutive dots and any character whose value is smaller than 32 or bigger 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 is used to delete a specific version of the specified file. If the Recycle Bin is enabled, the version is placed in the Recycle Bin instead. The operation MUST NOT grant deletion of the current version and the published version [<1>](#) of the file.

The following example shows this operation.

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

If versioning is disabled, the DeleteVersion operation MUST return error 0x80131600 indicating that the version does not exist.

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 WSDL message definitions are specific to this 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 defined as follows.

```
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 action value of the message is defined as follows.

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

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

3.1.4.2.2 Elements

The following XML Schema (2) element definitions are specific to this operation.

3.1.4.2.2.1 DeleteVersion

The definition of the **DeleteVersion** element is as follows.

```
<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: For information about this element, see section [2.2.3.1](#).

fileVersion: For information about this element, see section [2.2.3.2](#).

3.1.4.2.2.2 DeleteVersionResponse

The definition of the **DeleteVersionResponse** element is as follows.

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

DeleteVersionResult: Must return an XML node that conforms to the structure specified in section [2.2.4.1](#), that contains the details about all the versions of the specified file that the user can access.

Possible specific error conditions encountered during this operation generate the error codes shown in the following table.

Error code	Specification
	The fileName sub element in the DeleteVersion element contains invalid characters. Invalid characters defined in code are " # % & * : < = > ? \ { } ~ two consecutive dots and any character whose value is smaller than 32 or bigger than 126 <2> .
0x81020073	The fileName sub element in the DeleteVersion element contains invalid characters. Invalid characters defined in code are " # % & * : < = > ? \ { } ~ two consecutive dots and any character whose value is smaller than 32 or bigger than 126 <3> .
0x81070970	The fileName sub element in the DeleteVersion element contains invalid characters. Invalid characters defined in code are " # % & * : < = > ? \ { } ~ two consecutive dots and any character whose value is smaller than 32 or bigger than 126 <4> .
0x80131600	The specified version does not exist or was recently deleted. This error code is returned when the value specified by the fileVersion element is the current or published version of the file or the file does not exist. <5>

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 is used to get details about all versions of the specified file that the user can access. The following example shows this operation.

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

```
<wsdl:output message="tns:GetVersionsSoapOut" />
</wsdl:operation>
```

If versioning is disabled, only information about one version will be returned. This version MUST be the current version.

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 WSDL message definitions are specific to this 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 defined as follows.

```
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 action value of the message is defined as follows.

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

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

3.1.4.3.2 Elements

The following XML Schema (2) element definitions are specific to this operation.

3.1.4.3.2.1 GetVersions

The definition of the **GetVersions** element is as follows.

```
<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: For information about this element, see section [2.2.3.1](#).

3.1.4.3.2 GetVersionsResponse

The definition of the **GetVersionsResponse** element is as follows.

```
<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: MUST return an XML node that conforms to the structure specified in section [2.2.4.1](#), that contains the details about all the versions of the specified file that the user can access.

Specific error conditions encountered during this operation generate the error code shown in the following table.

Error code	Specification
	The system cannot find the file specified. 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 is used to restore the specified file to a specific version. After restoring to the old data, the current version number of the file MUST still be increased, as with any other change.

When restoring a version, the version MUST NOT be in the Recycle Bin and it 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, the operation MUST NOT allow restoration of the file.

If the protocol server enforces that only checked-out files can be modified, the caller calls the **CheckOutFile** operation, as specified in [\[MS-LISTSWS\]](#) section 3.1.4.8, before invoking this operation.

After the operation, the caller either calls the **CheckInFile** operation, as specified in [\[MS-LISTSWS\]](#) section 3.1.4.7, to commit the changes done by this operation or calls the **UndoCheckOut** operation, as specified in [\[MS-LISTSWS\]](#) section 3.1.4.26, to revert the changes done by this operation.

If check-out is not being enforced, there is no need to check out the file before restore, and there is no need to check in the file after restore. After restoring to the old data, if the file is not checked out, the current version number of the file MUST still be increased, as with any other change. If the file is checked out, the version number of the file MUST remain the same.

If versioning is disabled, the **RestoreVersion** operation MUST return an error indicating a bad URL.

The following example shows the **RestoreVersion** operation.

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

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 WSDL message definitions are specific to this 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 defined as follows.

```
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 action value of the message is defined as follows.

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

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

3.1.4.4.2 Elements

The following XML Schema (2) element definitions are specific to this operation.

3.1.4.4.2.1 RestoreVersion

The definition of the **RestoreVersion** element is as follows.

```
<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: For information about this element, see section [2.2.3.1](#).

fileVersion: For information about this element, see section [2.2.3.2](#).

3.1.4.4.2.2 RestoreVersionResponse

The definition of the **RestoreVersionResponse** element is as follows.

```
<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: An XML node that contains details about all the versions of the specified file that the user can access. The **Results** complex type defines the structure of the XML node.

Possible error conditions encountered during this operation generate the error codes shown in the following table.

Error code	Specification
0x8007009e	The file is not checked out. First check out this document before making changes.

Error code	Specification
0x81070975	The file is not checked out. The protocol server requires the file to be checked out before performing the RestoreVersion operation.
0x81070906	The file specified by fileName and fileVersion sub elements in the Restore Version element could not be found in the site (2).
	There are no files with the URL fileName in this web.
	The file or folder name fileName contains invalid characters. Please use a different name. Common invalid characters include the following: # % & * : < > ? / { } <6> . In this case, "~" and "." are also invalid characters, but "#" is not.
0x81070970	The file or folder name fileName contains invalid characters. Please use a different name. Common invalid characters include the following: # % & * : < > ? / { } <7> . In this case, "~" and "." are also invalid characters, but "#" is not.
0x81020073	The file or folder name fileName contains invalid characters. Please use a different name. Common invalid characters include the following: # % & * : < > ? / { } <8> . In this case, "~" and "." are also invalid characters, but "#" is not.

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

4.1 Restoring a File to Its First Version

In this scenario, a site (2) called TestSite contains a folder hierarchy of the form RootFolder/SubFolder on the protocol server. Three versions of a file called Test.txt exist in the folder named SubFolder. The versions are 0.1, 0.2, and 0.3; 0.1 is the first version created, and 0.3 is the current version. The protocol client has to restore the file to version 0.1.

1. To obtain the current list of versions for the file Test.txt, the protocol client issues a **GetVersions** request to TestSite, specifying the site (2)-relative file location (that is, RootFolder/SubFolder/Test.txt).

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

2. The protocol server returns a list of versions of the file 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" />
    </GetVersionsResult>
  </GetVersionsResponse>
</results>
```

3. The protocol client parses the response and finds the first version of the file, 0.1.
4. The protocol client issues a **CheckOutFile** request, as described in [\[MS-LISTSWS\]](#) section 3.1.4.8, giving 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>
```

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

6. 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>
```

7. The protocol server now has four versions of the file, namely "0.1", "0.2", "0.3", and "0.4". The contents of files with version "0.4" and "0.1" are identical.

5 Security

5.1 Security Considerations for Implementers

This protocol introduces no additional security considerations beyond those applicable to its underlying protocols.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the following full WSDL is provided.

```
<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:operation>
  </wsdl:binding>

```

```

        </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® SharePoint® Foundation 2010
- Windows® SharePoint® Services 2.0
- Windows® SharePoint® Services 3.0

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 3.1.4.2:](#) If minor version is enabled, the user interface version number for a published version is incremented to the next positive major version number and the minor version is 0. If minor version is not enabled, published version is the version which passes content approval. Each file can have at most one published version.

[<2> Section 3.1.4.2.2.2:](#) Windows SharePoint Services truncates all characters after the first "#" in the file name. If the truncated file name does not exist in server, 0x80131600 would be returned.

[<3> Section 3.1.4.2.2.2:](#) Windows SharePoint Services truncates all characters after the first "#" in the file name. If the truncated file name does not exist in server, 0x80131600 would be returned.

[<4> Section 3.1.4.2.2.2:](#) Windows SharePoint Services truncates all characters after the first "#" in the file name. If the truncated file name does not exist in server, 0x80131600 would be returned.

[<5> Section 3.1.4.2.2.2:](#) Windows SharePoint Services returns an error string of "Object reference not set to an instance of an object." without error code element when fileVersion is the current version of the file.

[<6> Section 3.1.4.4.2.2:](#) Windows SharePoint Services returns an error code "0x81070970" with the description "The file or folder name **filename** contains invalid characters. Please use a different name. Common invalid characters include the following: # % & * : < > ? / { | }".

[<7> Section 3.1.4.4.2.2:](#) Windows SharePoint Services returns an error code "0x81070970" with the description "The file or folder name **filename** contains invalid characters. Please use a different name. Common invalid characters include the following: # % & * : < > ? / { | }".

[<8> Section 3.1.4.4.2.2:](#) Windows SharePoint Services returns an error code "0x81070970" 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

A

Abstract data model
 [server](#) 14
[Applicability](#) 7
[Attribute groups](#) 13
[Attributes](#) 13

C

[Capability negotiation](#) 8
[Change tracking](#) 34
Client
 [overview](#) 14
[Complex types](#) 10
 [Results](#) 11
 [SOAPFaultDetails](#) 11
 [VersionData](#) 12

D

Data model - abstract
 [server](#) 14

E

Elements
 [fileName](#) 10
 [fileVersion](#) 10
Events
 [local - server](#) 24
 [timer - server](#) 24
Examples
 [restoring file to its first version](#) 25

F

[Fields - vendor-extensible](#) 8
[fileName element](#) 10
[fileVersion element](#) 10
[Full WSDL](#) 28

G

[Glossary](#) 5
[Groups](#) 13

I

[Implementer - security considerations](#) 27
[Index of security parameters](#) 27
[Informative references](#) 6
Initialization
 [server](#) 14
[Introduction](#) 5

L

Local events

[server](#) 24

M

Message processing
 [server](#) 14
Messages
 [attribute groups](#) 13
 [attributes](#) 13
 [complex types](#) 10
 [elements](#) 10
 [enumerated](#) 9
 [fileName element](#) 10
 [fileVersion element](#) 10
 [groups](#) 13
 [namespaces](#) 9
 [Results complex type](#) 11
 [simple types](#) 12
 [SOAPFaultDetails complex type](#) 11
 [syntax](#) 9
 [transport](#) 9
 [VersionData complex type](#) 12

N

[Namespaces](#) 9
[Normative references](#) 5

O

Operations
 [DeleteAllVersions](#) 15
 [DeleteVersion](#) 17
 [GetVersions](#) 19
 [RestoreVersion](#) 21
 [Overview \(synopsis\)](#) 6

P

[Parameters - security index](#) 27
[Preconditions](#) 7
[Prerequisites](#) 7
[Product behavior](#) 33

R

References
 [informative](#) 6
 [normative](#) 5
[Relationship to other protocols](#) 7
[Restoring file to its first version example](#) 25
[Results complex type](#) 11

S

Security
 [implementer considerations](#) 27
 [parameter index](#) 27
Sequencing rules

- [server](#) 14
- Server
 - [abstract data model](#) 14
 - [DeleteAllVersions operation](#) 15
 - [DeleteVersion operation](#) 17
 - [details](#) 14
 - [GetVersions operation](#) 19
 - [initialization](#) 14
 - [local events](#) 24
 - [message processing](#) 14
 - [overview](#) 14
 - [RestoreVersion operation](#) 21
 - [sequencing rules](#) 14
 - [timer events](#) 24
 - [timers](#) 14
- [Simple types](#) 12
- [SOAPFaultDetails complex type](#) 11
- [Standards assignments](#) 8
- Syntax
 - [messages - overview](#) 9

T

- Timer events
 - [server](#) 24
- Timers
 - [server](#) 14
- [Tracking changes](#) 34
- [Transport](#) 9
- Types
 - [complex](#) 10
 - [simple](#) 12

V

- [Vendor-extensible fields](#) 8
- [VersionData complex type](#) 12
- [Versioning](#) 8

W

- [WSDL](#) 28