

[MS-OFFICIALFILE]: Official File 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 [Open Specification Promise](#) or the [Community Promise](#). 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
04/25/2008	0.2	Editorial	Revised and edited the technical content
06/27/2008	1.0	Major	Revised and edited the technical content
01/16/2009	1.01	Editorial	Revised and edited the technical content
07/13/2009	1.02	Major	Revised and edited the technical content
08/28/2009	1.03	Editorial	Revised and edited the technical content
11/06/2009	1.04	Editorial	Revised and edited the technical content
02/19/2010	2.0	Minor	Updated 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	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.05	Editorial	Changed language and formatting in the technical content.
12/17/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.6	Minor	Clarified the meaning of the technical content.
04/11/2012	2.6	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.6	No change	No changes to the meaning, language, or formatting of the technical content.
09/12/2012	2.6	No change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
10/08/2012	2.7	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction	7
1.1 Glossary	7
1.2 References	7
1.2.1 Normative References	8
1.2.2 Informative References	8
1.3 Overview	8
1.4 Relationship to Other Protocols	9
1.5 Prerequisites/Preconditions	9
1.6 Applicability Statement	9
1.7 Versioning and Capability Negotiation	9
1.8 Vendor-Extensible Fields	9
1.9 Standards Assignments	9
2 Messages	10
2.1 Transport	10
2.2 Common Message Syntax	10
2.2.1 Namespaces	10
2.2.2 Messages	10
2.2.3 Elements	11
2.2.4 Complex Types	11
2.2.4.1 ArrayOfRecordsRepositoryProperty	11
2.2.4.2 CustomProcessingResult	11
2.2.4.3 RecordsRepositoryProperty	11
2.2.4.4 ServerInfo	12
2.2.4.5 SubmitFileResult	13
2.2.5 Simple Types	13
2.2.5.1 HoldProcessingResult	14
2.2.5.2 SubmitFileResultCode	14
2.2.6 Attributes	15
2.2.7 Groups	15
2.2.8 Attribute Groups	15
2.2.9 Common Data Structures	15
3 Protocol Details	16
3.1 Official File Server Details	16
3.1.1 Abstract Data Model	16
3.1.2 Timers	16
3.1.3 Initialization	17
3.1.4 Message Processing Events and Sequencing Rules	17
3.1.4.1 GetFinalRoutingDestinationFolderUrl	17
3.1.4.1.1 Messages	18
3.1.4.1.1.1 GetFinalRoutingDestinationFolderUrlSoapIn	18
3.1.4.1.1.2 GetFinalRoutingDestinationFolderUrlSoapOut	19
3.1.4.1.2 Elements	19
3.1.4.1.2.1 GetFinalRoutingDestinationFolderUrl	19
3.1.4.1.2.2 GetFinalRoutingDestinationFolderUrlResponse	19
3.1.4.1.3 Complex Types	20
3.1.4.1.3.1 DocumentRoutingResult	20
3.1.4.1.4 Simple Types	20
3.1.4.1.4.1 DocumentRoutingResultType	21

3.1.4.1.4.2 DocumentRoutingCollisionSetting	22
3.1.4.1.5 Attributes.....	22
3.1.4.1.6 Groups.....	22
3.1.4.1.7 Attribute Groups	22
3.1.4.2 GetHoldsInfo.....	22
3.1.4.2.1 Messages	23
3.1.4.2.1.1 GetHoldsInfoSoapIn.....	23
3.1.4.2.1.2 GetHoldsInfoSoapOut.....	23
3.1.4.2.2 Elements.....	23
3.1.4.2.2.1 GetHoldsInfo.....	23
3.1.4.2.2.2 GetHoldsInfoResponse	23
3.1.4.2.3 Complex Types	24
3.1.4.2.3.1 ArrayOfHoldInfo	24
3.1.4.2.3.2 HoldInfo	24
3.1.4.2.4 Simple Types.....	25
3.1.4.2.4.1 guid.....	25
3.1.4.2.5 Attributes.....	25
3.1.4.2.6 Groups.....	25
3.1.4.2.7 Attribute Groups	25
3.1.4.3 GetRecordRouting.....	26
3.1.4.3.1 Messages	26
3.1.4.3.1.1 GetRecordRoutingSoapIn.....	26
3.1.4.3.1.2 GetRecordRoutingSoapOut.....	26
3.1.4.3.2 Elements.....	26
3.1.4.3.2.1 GetRecordRouting.....	27
3.1.4.3.2.2 GetRecordRoutingResponse	27
3.1.4.3.3 Complex Types	27
3.1.4.3.4 Simple Types.....	27
3.1.4.3.5 Attributes.....	27
3.1.4.3.6 Groups.....	27
3.1.4.3.7 Attribute Groups	27
3.1.4.4 GetRecordRoutingCollection	27
3.1.4.4.1 Messages	28
3.1.4.4.1.1 GetRecordRoutingCollectionSoapIn	28
3.1.4.4.1.2 GetRecordRoutingCollectionSoapOut	28
3.1.4.4.2 Elements.....	28
3.1.4.4.2.1 GetRecordRoutingCollection	29
3.1.4.4.2.2 GetRecordRoutingCollectionResponse.....	29
3.1.4.4.3 Complex Types	29
3.1.4.4.4 Simple Types.....	29
3.1.4.4.5 Attributes.....	29
3.1.4.4.6 Groups.....	29
3.1.4.4.7 Attribute Groups	29
3.1.4.5 GetServerInfo	29
3.1.4.5.1 Messages	30
3.1.4.5.1.1 GetServerInfoSoapIn	30
3.1.4.5.1.2 GetServerInfoSoapOut	30
3.1.4.5.2 Elements.....	30
3.1.4.5.2.1 GetServerInfo	30
3.1.4.5.2.2 GetServerInfoResponse.....	31
3.1.4.5.3 Complex Types	31
3.1.4.5.4 Simple Types.....	31
3.1.4.5.5 Attributes.....	31

3.1.4.5.6	Groups.....	31
3.1.4.5.7	Attribute Groups	31
3.1.4.6	SubmitFile	31
3.1.4.6.1	Messages	33
3.1.4.6.1.1	SubmitFileSoapIn	33
3.1.4.6.1.2	SubmitFileSoapOut	33
3.1.4.6.2	Elements.....	33
3.1.4.6.2.1	SubmitFile	33
3.1.4.6.2.2	SubmitFileResponse.....	34
3.1.4.6.3	Complex Types	34
3.1.4.6.4	Simple Types.....	34
3.1.4.6.5	Attributes.....	34
3.1.4.6.6	Groups.....	35
3.1.4.6.7	Attribute Groups	35
3.1.5	Timer Events	35
3.1.6	Other Local Events	35
4	Protocol Examples.....	36
5	Security.....	42
5.1	Security Considerations for Implementers.....	42
5.2	Index of Security Parameters	42
6	Appendix A: Full WSDL	43
7	Appendix B: Full XML Schema	51
8	Appendix C: Product Behavior	52
9	Change Tracking.....	54
10	Index	56

1 Introduction

This document specifies the Official File Web Service Protocol, which enables the submission of files to a repository.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

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

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

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

audit log
base64 encoding
data type
document set
field
legal hold
login name
Simple Object Access Protocol (SOAP)
site
SOAP action
SOAP body
SOAP fault
Uniform Resource Locator (URL)
URL encode
Web Services Description Language (WSDL)
WSDL message
WSDL operation
XML namespace
XML namespace prefix
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

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

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

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989, <http://www.ietf.org/rfc/rfc1123.txt>

[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.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] 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.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](#)".

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

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

1.3 Overview

This protocol enables a protocol client to submit a file and its associated properties to a repository for storage. It also provides the ability to retrieve data about the repository, such as its type, version, and the rules that determine the appropriate storage location for each file type.

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

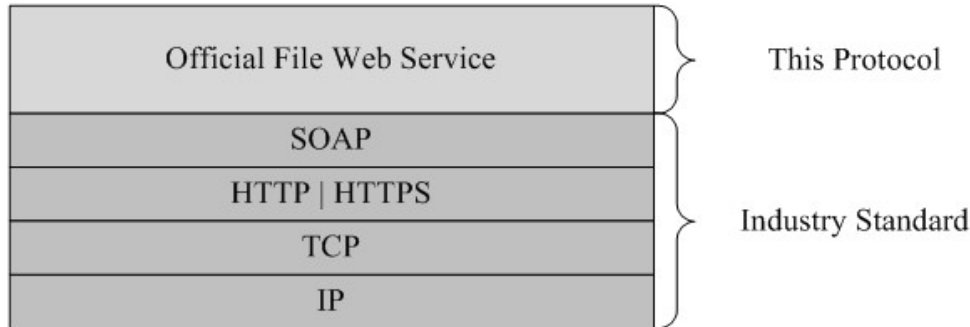


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a repository that is identified by a **URL** that is known by protocol clients. The protocol server endpoint is formed by appending `"/_vti_bin/OfficialFile.asmx "` to the URL of the repository, for example, `http://www.contoso.com/_vti_bin/OfficialFile.asmx`.

This protocol assumes that authentication has been performed by the underlying protocols.

1.6 Applicability Statement

This protocol is designed for submission of files less than 50 megabytes in size to the protocol server.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

Supported Transports: This protocol uses multiple transports with SOAP as described in section [2.1](#).

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

In the following sections, the schema definition might be less restrictive than the processing rules imposed by the protocol. The **WSDL** in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL specifies additional restrictions that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be empty, null, or not present but the behavior of the protocol as specified restricts the same elements to being non-empty, not null, and present.

2.1 Transport

Protocol servers **MUST** support SOAP over HTTP. Protocol servers **SHOULD** additionally support SOAP over HTTPS for securing communication with 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 using HTTP status codes as specified in [\[RFC2616\]](#) (Section 10, Status Code Definitions) or using **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 definitions that are used by this protocol. The syntax of the definitions uses **XML schema**, as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and WSDL, as specified in [\[WSDL\]](#).

2.2.1 Namespaces

This specification defines and references various **XML namespaces** 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/	
s1	http://microsoft.com/wsdl/types/	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
tns	http://schemas.microsoft.com/sharepoint/soap/recordsrepository/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

2.2.2 Messages

This specification does not define any common **WSDL message** definitions.

2.2.3 Elements

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

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
ArrayOfRecordsRepositoryProperty	A list of properties associated with the file being submitted.
CustomProcessingResult	The result of custom processing of a legal hold .
RecordsRepositoryProperty	A property of the file being submitted.
ServerInfo	Server information.
SubmitFileResult	The detailed data result for the SubmitFile WSDL operation.

2.2.4.1 ArrayOfRecordsRepositoryProperty

Namespace: <http://schemas.microsoft.com/sharepoint/soap/recordsrepository/>

A list of properties associated with the file being submitted.

```
<xs:complexType name="ArrayOfRecordsRepositoryProperty"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="RecordsRepositoryProperty"
nillable="true" type="tns:RecordsRepositoryProperty"/>
  </xs:sequence>
</xs:complexType>
```

RecordsRepositoryProperty: A property of the file being submitted.

2.2.4.2 CustomProcessingResult

Namespace: <http://schemas.microsoft.com/sharepoint/soap/recordsrepository/>

The result of custom processing of a legal hold.

```
<xs:complexType name="CustomProcessingResult" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="HoldsProcessingResult" type="tns:HoldProcessingResult"/>
  </xs:sequence>
</xs:complexType>
```

HoldsProcessingResult: The result of processing a legal hold.

2.2.4.3 RecordsRepositoryProperty

Namespace: <http://schemas.microsoft.com/sharepoint/soap/recordsrepository/>

A property of the file being submitted.

```
<xs:complexType name="RecordsRepositoryProperty" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DisplayName" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Value" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Type" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Other" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Name: The name of the property, which MUST be non-empty and MUST be less than or equal to 256 characters in length. The properties described in the following table are common.

Value	Meaning
_dlc_hold_url	The URL of the location where a legal hold is defined. Type MUST be set to "OfficialFileCustomType".
_dlc_hold_id	The identifier of a legal hold, which MUST be a positive integer. Type MUST be set to "OfficialFileCustomType".
_dlc_hold_comments	A human readable comment associated with the legal hold. Type MUST be set to "OfficialFileCustomType".
_dlc_hold_searchquery	A search query associated with the legal hold. Type MUST be set to "OfficialFileCustomType".
_dlc_hold_searchcontexturl	The implementation-specific<1> identifier of the location where the search query associated with the legal hold was performed. Type MUST be set to "OfficialFileCustomType".
AuditHistory	The implementation-specific audit log for the file. Type MUST be set to XML.

DisplayName: Display name of the property. This element SHOULD<2> be present. If present, it MUST be non-empty and MUST be less than or equal to 256 characters in length.

Value: The value of the property, which is a value appropriate for the **Type** as specified for this property or a string if **Type** is "OfficialFileCustomType". If **Type** is DateTime, then the value SHOULD<3> be formatted as specified in [RFC1123].

Type: The type of the property, which MUST be a valid **data type**, a value from the **Field Types** as specified in [MS-WSSTS] section 2.3 or XML or "OfficialFileCustomType".

Other: Other related data. If **Name** is set to "AuditHistory", this is the implementation-specific audit log for the file. Otherwise, this element is reserved and MUST be empty.

2.2.4.4 ServerInfo

Namespace: http://schemas.microsoft.com/sharepoint/soap/recordsrepository/

Server information.

```
<xs:complexType name="ServerInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
```

```

    <xs:element name="ServerType" type="xs:string"/>
    <xs:element name="ServerVersion" type="xs:string"/>
    <xs:element name="RoutingWeb" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

ServerType: The type information for the protocol server. MUST be non-empty and MUST be less than or equal to 256 characters in length and MUST be HTML encoded.

ServerVersion: The version information for the protocol server. MUST be non-empty and MUST be less than or equal to 256 characters in length and MUST be HTML encoded.

RoutingWeb: Whether the protocol server is configured for routing. This element SHOULD [<4>](#) be present. If present, it MUST conform to the XML schema of the **BooleanType** simple type.

2.2.4.5 SubmitFileResult

Namespace: http://schemas.microsoft.com/sharepoint/soap/recordsrepository/

The detailed data result for the **SubmitFile** WSDL operation.

```

<xs:complexType name="SubmitFileResult" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element name="ResultCode" type="tns:SubmitFileResultCode"/>
    <xs:choice>
      <xs:element minOccurs="0" name="ResultUrl" type="xs:anyURI"/>
      <xs:element minOccurs="0" name="AdditionalInformation" type="xs:string"/>
    </xs:choice>
    <xs:element minOccurs="0" name="CustomProcessingResult"
      type="tns:CustomProcessingResult"/>
  </xs:sequence>
</xs:complexType>

```

ResultCode: The result of the **SubmitFile** WSDL operation.

ResultUrl: The URL of the file after the **SubmitFile** WSDL operation.

AdditionalInformation: Additional information specific to the **ResultCode** element value as specified in section [3.1.4.6](#).

CustomProcessingResult: The result of custom file processing.

2.2.5 Simple Types

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

Simple type	Description
HoldProcessingResult	The result of processing a legal hold.
SubmitFileResultCode	The result status code of a SubmitFile WSDL operation.

2.2.5.1 HoldProcessingResult

Namespace: http://schemas.microsoft.com/sharepoint/soap/recordsrepository/

The result of processing a legal hold.

```
<xs:simpleType name="HoldProcessingResult" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Success"/>
    <xs:enumeration value="Failure"/>
    <xs:enumeration value="InDropOffZone"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **HoldProcessingResult** simple type.

Value	Meaning
Success	The processing of a legal hold is successful.
Failure	The processing of a legal hold failed.
InDropOffZone	

2.2.5.2 SubmitFileResultCode

Namespace: http://schemas.microsoft.com/sharepoint/soap/recordsrepository/

The result status code of a **SubmitFile** WSDL operation.

```
<xs:simpleType name="SubmitFileResultCode" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Success"/>
    <xs:enumeration value="MoreInformation"/>
    <xs:enumeration value="InvalidRouterConfiguration"/>
    <xs:enumeration value="InvalidArgument"/>
    <xs:enumeration value="InvalidUser"/>
    <xs:enumeration value="NotFound"/>
    <xs:enumeration value="FileRejected"/>
    <xs:enumeration value="UnknownError"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **SubmitFileResultCode** simple type.

Value	Meaning
Success	The operation is successful.
MoreInformation	The operation is successful but further action is needed.
InvalidRouterConfiguration	The operation failed because the repository was not configured for routing.
InvalidArgument	The operation failed because of an invalid argument.

Value	Meaning
InvalidUser	The operation failed because of an invalid user.
NotFound	The operation failed because the user was not authorized to submit files.
FileRejected	The operation failed because of a rejected file.
UnknownError	The operation failed because of an unknown error.

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.

2.2.9 Common Data Structures

This specification does not define any common XML schema data structures.

3 Protocol Details

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The WSDL in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, and **present**.

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 notify protocol clients of application-level faults using SOAP faults. 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 using either HTTP status codes or SOAP faults, as specified previously in this section.

3.1 Official File Server Details

3.1.1 Abstract Data Model

This section specifies 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 document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The protocol server is a repository that maintains multiple storage locations where files and their associated properties can be stored. Storage locations can require a set of zero or more required properties that are stored with each file in that location. Certain of these storage locations can be configured to enforce that content submissions be routed by the protocol server. The protocol server maintains one storage location as a temporary storage location, for example, for content pending additional user interaction.

The protocol server maintains a list of users that are authorized to submit files to the repository.

The protocol server maintains a list of rules that specify file types and the location where files of those types are stored. There MUST be one or more rules, and exactly one rule MUST be configured as the default.

The protocol server maintains various configuration settings, such as whether file metadata is parsed from the contents of submitted files and what to do when a name collision occurs.

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
GetFinalRoutingDestinationFolderUrl	This method<5> is called to determine the storage location for the submission based on the rules in the repository and a suggested save location chosen by a user.
GetHoldsInfo	This method<6> is called to retrieve information about the legal holds in a repository.
GetRecordRouting	This method is deprecated and MUST NOT be called.
GetRecordRoutingCollection	This method is deprecated and MUST NOT be called.
GetServerInfo	This method is called to retrieve information about the repository, such as its type, version, and whether the repository is configured for routing.
SubmitFile	This method is called to submit a file and its associated properties to the repository.

3.1.4.1 GetFinalRoutingDestinationFolderUrl

This method<7> is called to determine the storage location for the submission based on the rules in the repository and a suggested save location chosen by a user.

```
<wsdl:operation name="GetFinalRoutingDestinationFolderUrl"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetFinalRoutingDestinationFolderUrlSoapIn"/>
  <wsdl:output message="tns:GetFinalRoutingDestinationFolderUrlSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetFinalRoutingDestinationFolderUrlSoapIn** request WSDL message, and the protocol server MUST respond with a **GetFinalRoutingDestinationFolderUrlSoapOut** response WSDL message, as follows:

1. If the repository is not configured for routing content then the protocol server MUST set the **ResultType** element to "DocumentRoutingDisabled" and return.
2. If the **originalSaveLocation** is a storage location for a **document set**, then the protocol server MUST set the **ResultType** element to "OriginalSaveLocationIsDocumentSet" and return.
3. If the **originalSaveLocation** is a storage location for which routing is not enforced, then the protocol server MUST set the **ResultType** element to "NoEnforcementAtOriginalSaveLocation" and return.
4. If the **contentTypeName** is NULL or empty then the protocol server MUST set the **ResultType** element to "MissingRequiredProperties" and return.

5. Next the protocol server processes the rules in an implementation-specific manner based on the submission's **properties**, **contentTypeName**, and **originalSaveLocation** to determine the storage location for the submission.
6. If no applicable rule is found then the protocol server MUST set the **ResultType** element to "SuccessToDropOffLibrary" and set the **Url** to the temporary storage location and return.
7. If the protocol server determines that the storage location determined by the rules has required properties that are not present in the **properties** element, the protocol server MUST set the **ResultType** element to "SuccessToDropOffLibrary", set the **Url** to the temporary storage location and return.
8. If the user does not have permissions to store content in the storage location determined by the rules, then the protocol server MUST set **ResultType** to "PermissionDeniedAtDestination" and **Url** to the temporary storage location and return.
9. If other implementation-specific errors occur while determining the storage location for the submission, the protocol server MUST set the **ResultType** element to "UnknownFailure" and return.
10. Otherwise, the protocol server sets **Url** to the storage location determined by the rules, which MUST be non-empty.
11. If parsing of file metadata is disabled on the repository, the protocol server MUST set the **ResultType** to "ParsingDisabledAtDestination". Otherwise the protocol server MUST set the **ResultType** to "Success".
12. If a file with the same name does not already exist at the location determined by the rule then the protocol server MUST set **CollisionSetting** to "NoCollision" and return. If the repository is configured to overwrite existing files and versioning is enabled at the location determined by the rule, then the protocol server MUST set **CollisionSetting** to "UseSharePointVersioning" and return. Otherwise the protocol server MUST set **CollisionSetting** to "AppendUniqueSuffixes".

3.1.4.1.1 Messages

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

Message	Description
GetFinalRoutingDestinationFolderUrlSoapIn	The request WSDL message for the GetFinalRoutingDestinationFolderUrl WSDL operation .
GetFinalRoutingDestinationFolderUrlSoapOut	The response WSDL message for the GetFinalRoutingDestinationFolderUrl WSDL operation .

3.1.4.1.1.1 GetFinalRoutingDestinationFolderUrlSoapIn

The request WSDL message for the **GetFinalRoutingDestinationFolderUrl WSDL operation**.

The **SOAP action** value is:

http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetFinalRoutingDestinationFolderUrl

The **SOAP body** contains the **GetFinalRoutingDestinationFolderUrl** element.

3.1.4.1.1.2 GetFinalRoutingDestinationFolderUrlSoapOut

The response WSDL message for the **GetFinalRoutingDestinationFolderUrl** WSDL operation.

The SOAP body contains the **GetFinalRoutingDestinationFolderUrlResponse** element.

3.1.4.1.2 Elements

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

Element	Description
GetFinalRoutingDestinationFolderUrl	The input data for the GetFinalRoutingDestinationFolderUrl WSDL operation.
GetFinalRoutingDestinationFolderUrlResponse	The result data for the GetFinalRoutingDestinationFolderUrl WSDL operation.

3.1.4.1.2.1 GetFinalRoutingDestinationFolderUrl

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

```
<xs:element name="GetFinalRoutingDestinationFolderUrl"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="properties"
type="tns:ArrayOfRecordsRepositoryProperty"/>
      <xs:element minOccurs="0" maxOccurs="1" name="contentTypeName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="originalSaveLocation" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

properties: The properties of the file.

contentTypeName: The file type, which MUST be less than or equal to 128 characters in length.

originalSaveLocation: The suggested save location chosen by a user, which MUST be a non-empty URL.

3.1.4.1.2.2 GetFinalRoutingDestinationFolderUrlResponse

The **GetFinalRoutingDestinationFolderUrlResponse** element specifies the result data for the **GetFinalRoutingDestinationFolderUrl** WSDL operation.

```

<xs:element name="GetFinalRoutingDestinationFolderUrlResponse"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1"
name="GetFinalRoutingDestinationFolderUrlResult" type="tns:DocumentRoutingResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

GetFinalRoutingDestinationFolderUrlResult: Data details about the result, which is an XML fragment that MUST conform to the XML schema of the **DocumentRoutingResult** complex type.

3.1.4.1.3 Complex Types

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

Complex type	Description
DocumentRoutingResult	Result data details for a GetFinalRoutingDestinationFolderUrl WSDL operation.

3.1.4.1.3.1 DocumentRoutingResult

Namespace: http://schemas.microsoft.com/sharepoint/soap/recordsrepository/

Result data details for a **GetFinalRoutingDestinationFolderUrl** WSDL operation.

```

<xs:complexType name="DocumentRoutingResult" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Url" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="ResultType"
type="tns:DocumentRoutingResultType"/>
    <xs:element minOccurs="1" maxOccurs="1" name="CollisionSetting"
type="tns:DocumentRoutingCollisionSetting"/>
  </xs:sequence>
</xs:complexType>

```

Url: An implementation-specific URL to the storage location for the submission.

ResultType: The result of the **GetFinalRoutingDestinationFolderUrl** WSDL operation.

CollisionSetting: Collision setting, that is, whether there was a file of the same name already stored at the storage location and, if so, the server configured behavior.

3.1.4.1.4 Simple Types

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

Simple type	Description
DocumentRoutingResultType	Type of result.

Simple type	Description
DocumentRoutingCollisionSetting	Collision setting for the repository, that is, whether there was a file of the same name already stored at the storage location and, if so, the server configured behavior.

3.1.4.1.4.1 DocumentRoutingResultType

Namespace: http://schemas.microsoft.com/sharepoint/soap/recordsrepository/

Type of result.

```
<xs:simpleType name="DocumentRoutingResultType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Success"/>
    <xs:enumeration value="SuccessToDropOffLibrary"/>
    <xs:enumeration value="MissingRequiredProperties"/>
    <xs:enumeration value="NoMatchingRules"/>
    <xs:enumeration value="DocumentRoutingDisabled"/>
    <xs:enumeration value="PermissionDeniedAtDestination"/>
    <xs:enumeration value="ParsingDisabledAtDestination"/>
    <xs:enumeration value="OriginalSaveLocationIsDocumentSet"/>
    <xs:enumeration value="NoEnforcementAtOriginalSaveLocation"/>
    <xs:enumeration value="UnknownFailure"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **DocumentRoutingResultType** simple type.

Value	Meaning
Success	The operation was successful.
SuccessToDropOffLibrary	The operation was successful and the location determined by the protocol server is the temporary storage location.
MissingRequiredProperties	Additional data required from the protocol client.
NoMatchingRules	Reserved. Protocol server MUST not return this result.
DocumentRoutingDisabled	Routing is not enabled on the repository
PermissionDeniedAtDestination	Implementation-specific error validating permissions for the user at the storage location.
ParsingDisabledAtDestination	Parsing is not enabled on the repository.
OriginalSaveLocationIsDocumentSet	The originalSaveLocation is a storage location for storing document sets.
NoEnforcementAtOriginalSaveLocation	The originalSaveLocation is a storage location for which routing is not enforced.
UnknownFailure	Generic error.

3.1.4.1.4.2 DocumentRoutingCollisionSetting

Namespace: http://schemas.microsoft.com/sharepoint/soap/recordsrepository/

Collision setting for the repository, that is, whether there was a file of the same name already stored at the storage location and, if so, the server configured behavior.

```
<xs:simpleType name="DocumentRoutingCollisionSetting"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NoCollision"/>
    <xs:enumeration value="UseSharePointVersioning"/>
    <xs:enumeration value="AppendUniqueSuffixes"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **DocumentRoutingCollisionSetting** simple type.

Value	Meaning
NoCollision	There was no file of the same name already stored at the storage location.
UseSharePointVersioning	Overwrite the existing file.
AppendUniqueSuffixes	Generate a unique file name.

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 GetHoldsInfo

This method [<8>](#) is called to retrieve information about the legal holds in a repository.

```
<wsdl:operation name="GetHoldsInfo" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetHoldsInfoSoapIn"/>
  <wsdl:output message="tns:GetHoldsInfoSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetHoldsInfoSoapIn** request WSDL message and the protocol server MUST respond with a **GetHoldsInfoSoapOut** response WSDL message as follows.

- The protocol server returns the data associated with the legal holds in the repository in the **GetHoldsInfoResult** element.

3.1.4.2.1 Messages

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

Message	Description
GetHoldsInfoSoapIn	The request WSDL message for the GetHoldsInfo WSDL operation.
GetHoldsInfoSoapOut	The response WSDL message for the GetHoldsInfo WSDL operation.

3.1.4.2.1.1 GetHoldsInfoSoapIn

The request WSDL message for the **GetHoldsInfo** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetHoldsInfo
```

The SOAP body contains the **GetHoldsInfo** element.

3.1.4.2.1.2 GetHoldsInfoSoapOut

The response WSDL message for the **GetHoldsInfo** WSDL operation.

The SOAP body contains the **GetHoldsInfoResponse** element.

3.1.4.2.2 Elements

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

Element	Description
GetHoldsInfo	The input data for the GetHoldsInfo WSDL operation.
GetHoldsInfoResponse	The result data for the GetHoldsInfo WSDL operation.

3.1.4.2.2.1 GetHoldsInfo

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

```
<xs:element name="GetHoldsInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:complexType/>  
</xs:element>
```

3.1.4.2.2.2 GetHoldsInfoResponse

The **GetHoldsInfoResponse** element specifies the result data for the **GetHoldsInfo** WSDL operation.

```
<xs:element name="GetHoldsInfoResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:complexType/>
```

```

    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetHoldsInfoResult"
type="tns:ArrayOfHoldInfo"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

GetHoldsInfoResult: A list of legal holds.

3.1.4.2.3 Complex Types

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

Complex type	Description
ArrayOfHoldInfo	A list of legal holds.
HoldInfo	A legal hold.

3.1.4.2.3.1 ArrayOfHoldInfo

Namespace: http://schemas.microsoft.com/sharepoint/soap/recordsrepository/

A list of legal holds.

```

<xs:complexType name="ArrayOfHoldInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="HoldInfo" nillable="true"
type="tns:HoldInfo"/>
  </xs:sequence>
</xs:complexType>

```

HoldInfo: A legal hold.

3.1.4.2.3.2 HoldInfo

Namespace: http://schemas.microsoft.com/sharepoint/soap/recordsrepository/

A legal hold.

```

<xs:complexType name="HoldInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Url" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Title" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Description" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ManagedBy" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="RepositoryName" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DiscoveryQueries" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Id" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="ListId"
xmlns:s1="http://microsoft.com/wsdl/types/" type="s1:guid"/>
    <xs:element minOccurs="1" maxOccurs="1" name="WebId"
xmlns:s1="http://microsoft.com/wsdl/types/" type="s1:guid"/>
  </xs:sequence>

```


</xs:complexType>

Url: URL of the legal hold, which MUST be non-empty.

Title: Title of the legal hold.

Description: Description of the legal hold.

ManagedBy: Name of the person managing the legal hold.

RepositoryName: Name of the container that contains the legal hold.

DiscoveryQueries: Textual search queries associated with the legal hold.

Id: Identifier of the legal hold, which MUST be a positive integer.

ListId: Identifier of the storage location of the legal hold, which MUST be a **GUID**.

WebId: Identifier of the repository that contains the legal hold, which MUST be a GUID.

3.1.4.2.4 Simple Types

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

Simple type	Description
guid	A GUID.

3.1.4.2.4.1 guid

Namespace: http://microsoft.com/wsd/types/

A GUID.

```
<xs:simpleType name="guid" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:restriction base="xs:string">  
    <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>  
  </xs:restriction>  
</xs:simpleType>
```

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 GetRecordRouting

This method is deprecated and MUST NOT be called.

```
<wsdl:operation name="GetRecordRouting" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetRecordRoutingSoapIn"/>
  <wsdl:output message="tns:GetRecordRoutingSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetRecordRoutingSoapIn** request WSDL message and the protocol server MUST respond with a **GetRecordRoutingSoapOut** response WSDL message, as follows:

The protocol server returns an implementation-specific value in the **GetRecordRoutingResult** element that MUST be ignored.

3.1.4.3.1 Messages

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

Message	Description
GetRecordRoutingSoapIn	The request WSDL message for the GetRecordRouting WSDL operation.
GetRecordRoutingSoapOut	The response WSDL message for the GetRecordRouting WSDL operation.

3.1.4.3.1.1 GetRecordRoutingSoapIn

The request WSDL message for the **GetRecordRouting** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetRecordRouting
```

The SOAP body contains the **GetRecordRouting** element.

3.1.4.3.1.2 GetRecordRoutingSoapOut

The response WSDL message for the **GetRecordRouting** WSDL operation.

The SOAP body contains the **GetRecordRoutingResponse** element.

3.1.4.3.2 Elements

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

Element	Description
GetRecordRouting	The input data for the GetRecordRouting WSDL operation.
GetRecordRoutingResponse	The result data for the GetRecordRouting WSDL operation.

3.1.4.3.2.1 GetRecordRouting

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

```
<xs:element name="GetRecordRouting" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="recordRouting" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

recordRouting: The file type.

3.1.4.3.2.2 GetRecordRoutingResponse

The **GetRecordRoutingResponse** element specifies the result data for the **GetRecordRouting** WSDL operation.

```
<xs:element name="GetRecordRoutingResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetRecordRoutingResult"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetRecordRoutingResult: Implementation-specific result data.

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 GetRecordRoutingCollection

This method is deprecated and MUST NOT be called.

```

<wsdl:operation name="GetRecordRoutingCollection"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetRecordRoutingCollectionSoapIn"/>
  <wsdl:output message="tns:GetRecordRoutingCollectionSoapOut"/>
</wsdl:operation>

```

The protocol client sends a **GetRecordRoutingCollectionSoapIn** request WSDL message, and the protocol server MUST respond with a **GetRecordRoutingCollectionSoapOut** response WSDL message, as follows:

- The protocol server returns an implementation-specific value in the **GetRecordRoutingCollectionResult** element that MUST be ignored.

3.1.4.4.1 Messages

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

Message	Description
GetRecordRoutingCollectionSoapIn	The request WSDL message for the GetRecordRoutingCollection WSDL operation.
GetRecordRoutingCollectionSoapOut	The response WSDL message for the GetRecordRoutingCollection WSDL operation.

3.1.4.4.1.1 GetRecordRoutingCollectionSoapIn

The request WSDL message for the **GetRecordRoutingCollection** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetRecordRoutingCollection
```

The SOAP body contains the **GetRecordRoutingCollection** element.

3.1.4.4.1.2 GetRecordRoutingCollectionSoapOut

The response WSDL message for the **GetRecordRoutingCollection** WSDL operation.

The SOAP body contains the **GetRecordRoutingCollectionResponse** element.

3.1.4.4.2 Elements

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

Element	Description
GetRecordRoutingCollection	The input data for the GetRecordRoutingCollection WSDL operation.
GetRecordRoutingCollectionResponse	The result data for the GetRecordRoutingCollection WSDL operation.

3.1.4.4.2.1 GetRecordRoutingCollection

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

```
<xs:element name="GetRecordRoutingCollection" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType/>
</xs:element>
```

3.1.4.4.2.2 GetRecordRoutingCollectionResponse

The **GetRecordRoutingCollectionResponse** element specifies the result data for the **GetRecordRoutingCollection** WSDL operation.

```
<xs:element name="GetRecordRoutingCollectionResponse"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetRecordRoutingCollectionResult"
        type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetRecordRoutingCollectionResult: Implementation-specific result data.

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.

3.1.4.5 GetServerInfo

This method is called to retrieve information about the repository, such as its type, version, and whether the repository is configured for routing.

```
<wsdl:operation name="GetServerInfo" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetServerInfoSoapIn"/>
```

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

The protocol client sends a **GetServerInfoSoapIn** request WSDL message, and the protocol server MUST respond with a **GetServerInfoSoapOut** response WSDL message, as follows:

- The protocol server MUST return implementation-specific [<9>](#) information for the protocol server in the **ServerInfo** element.

3.1.4.5.1 Messages

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

Message	Description
GetServerInfoSoapIn	The request WSDL message for the GetServerInfo WSDL operation.
GetServerInfoSoapOut	The response WSDL message for the GetServerInfo WSDL operation.

3.1.4.5.1.1 GetServerInfoSoapIn

The request WSDL message for the **GetServerInfo** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetServerInfo
```

The SOAP body contains the **GetServerInfo** element.

3.1.4.5.1.2 GetServerInfoSoapOut

The response WSDL message for the **GetServerInfo** WSDL operation.

The SOAP body contains the **GetServerInfoResponse** element.

3.1.4.5.2 Elements

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

Element	Description
GetServerInfo	The input data for the GetServerInfo WSDL operation.
GetServerInfoResponse	The result data for the GetServerInfo WSDL operation.

3.1.4.5.2.1 GetServerInfo

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

```
<xs:element name="GetServerInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType/>
```

```
</xs:element>
```

3.1.4.5.2 GetServerInfoResponse

The **GetServerInfoResponse** element specifies the result data for the **GetServerInfo** WSDL operation.

```
<xs:element name="GetServerInfoResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element minOccurs="0" maxOccurs="1" name="GetServerInfoResult" type="xs:string"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

GetServerInfoResult: Type and version information for a protocol server, which MUST be an XML fragment that conforms to the XML schema of the **ServerInfo** complex type.

3.1.4.5.3 Complex Types

None.

3.1.4.5.4 Simple Types

None.

3.1.4.5.5 Attributes

None.

3.1.4.5.6 Groups

None.

3.1.4.5.7 Attribute Groups

None.

3.1.4.6 SubmitFile

This method is called to submit a file and its associated properties to the repository.

```
<wsdl:operation name="SubmitFile" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">  
  <wsdl:input message="tns:SubmitFileSoapIn"/>  
  <wsdl:output message="tns:SubmitFileSoapOut"/>  
</wsdl:operation>
```

The protocol client sends a **SubmitFileSoapIn** request WSDL message, and the protocol server MUST respond with a **SubmitFileSoapOut** response WSDL message, as follows:

1. If the user is not in the list of users authorized to submit files to the repository, the protocol server MUST set the **ResultCode** element to "NotFound" and return.

2. If required parameters were not specified, the protocol server MUST set the **ResultCode** to InvalidArgument and return.
3. If the protocol server determines that the file cannot be submitted to the repository using an implementation-specific<10> algorithm, the protocol server MUST set the **ResultCode** element to FileRejected and return.
4. If the protocol server determines that the repository is not configured for routing, the protocol server MUST set the **ResultCode** element to "InvalidRouterConfiguration" and return.
5. The protocol server processes the rules in an implementation-specific manner based on the submission's **properties** and **recordRouting** to determine the storage location for the submission.
6. If no applicable rule is found:
 1. The protocol server sets the **ResultCode** element to MoreInformation.
 2. The protocol server MUST set the **ResultUrl** element to an implementation-specific URL to enter more information about the submission.
 3. The protocol server SHOULD<11> set the **CustomProcessingResult.HoldsProcessingResult** element to InDropOffZone.
 4. The protocol server uses the temporary storage location.
7. If the protocol server determines that the storage location determined by the rules has required properties that are not present in the **properties** element:
 1. If the protocol server determines that the name of the user specified in the **userName** element is invalid using an implementation-specific validation algorithm, then the protocol server MUST set the **ResultCode** element to InvalidUser and return.
 2. Otherwise, the protocol server MUST set the **ResultCode** element to MoreInformation and the **ResultUrl** element to an implementation-specific URL to enter more information about the submission and SHOULD<12> set the **CustomProcessingResult.HoldsProcessingResult** element to InDropOffZone. The protocol server uses the temporary storage location.
8. The protocol server stores the file in the storage location.
9. If implementation-specific errors occur while determining the storage location for the submission or while storing the file, the protocol server MUST set the **ResultCode** element to UnknownError and return.
10. Otherwise, the protocol server MUST set the **ResultCode** element to Success and SHOULD<13> set the **ResultUrl** element to a non-empty HTML encoded URL to retrieve the stored file.
11. If the file has been stored in the temporary storage location, the server returns.
12. If the **properties** element contains all of the following properties, then the protocol server sets the implementation-specific metadata that indicates the file is placed on the appropriate legal hold. The protocol server SHOULD<14> set the **CustomProcessingResult.HoldsProcessingResult** element to Success.
 - _dlc_hold_url
 - _dlc_hold_comments

- _dlc_hold_id
- _dlc_hold_searchquery
- _dlc_hold_searchcontexturl

13.If, however, at least one but not all of the properties mentioned previously are contained in the **properties** element, then the protocol server SHOULD [<15>](#) set the **CustomProcessingResult.HoldsProcessingResult** element to Failure.

14.If none of the properties mentioned previously are contained in the **properties** element then the protocol server MUST NOT set **CustomProcessingResult.HoldsProcessingResult**.

3.1.4.6.1 Messages

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

Message	Description
SubmitFileSoapIn	The request WSDL message for the SubmitFile WSDL operation.
SubmitFileSoapOut	The response WSDL message for the SubmitFile WSDL operation.

3.1.4.6.1.1 SubmitFileSoapIn

The request WSDL message for the **SubmitFile** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/soap/recordsrepository/SubmitFile
```

The SOAP body contains the **SubmitFile** element.

3.1.4.6.1.2 SubmitFileSoapOut

The response WSDL message for the **SubmitFile** WSDL operation.

The SOAP body contains the **SubmitFileResponse** element.

3.1.4.6.2 Elements

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

Element	Description
SubmitFile	The input data for the SubmitFile WSDL operation.
SubmitFileResponse	The result data for the SubmitFile WSDL operation.

3.1.4.6.2.1 SubmitFile

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

```

<xs:element name="SubmitFile" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="fileToSubmit" type="xs:base64Binary"/>
      <xs:element minOccurs="0" maxOccurs="1" name="properties"
type="tns:ArrayOfRecordsRepositoryProperty"/>
      <xs:element minOccurs="0" maxOccurs="1" name="recordRouting" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sourceUrl" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="userName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

fileToSubmit: The contents of the file, which MUST be non-empty and MUST be encoded with **base64 encoding**.

properties: The properties of the file. MUST not be null.

recordRouting: The file type, which MUST be less than or equal to 128 characters in length. MUST not be null or empty.

sourceUrl: The source URL of the file, which MUST be non-empty and MUST be **URL-encoded**.

userName: The name of the user submitting the file, which MUST be non-empty and MUST be a **login name**. The value SHAREPOINT\system (case-insensitive) is reserved and SHOULD NOT [<16>](#) be used.

3.1.4.6.2 SubmitFileResponse

The **SubmitFileResponse** element specifies the result data for the **SubmitFile** WSDL operation.

```

<xs:element name="SubmitFileResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="SubmitFileResult" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

SubmitFileResult: Data details about the result, which is a string of an encoded XML fragment that MUST conform to the XML schema of the **SubmitFileResult** complex type.

3.1.4.6.3 Complex Types

None.

3.1.4.6.4 Simple Types

None.

3.1.4.6.5 Attributes

None.

3.1.4.6.6 Groups

None.

3.1.4.6.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

A protocol client might construct the following WSDL message to find the exact final destination location for file with three properties that is intended to be stored at "http://www.contoso.com/blue/Blue-Car.jpg" and of file type "Document".

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetFinalRoutingDestinationFolderUrl
xmlns="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/">
      <properties>
        <RecordsRepositoryProperty>
          <Name>Color</Name>
          <DisplayName>Color</DisplayName>
          <Value>blue</Value>
          <Type>Text</Type>
          <Other />
        </RecordsRepositoryProperty>
        <RecordsRepositoryProperty>
          <Name>Title</Name>
          <DisplayName>Title</DisplayName>
          <Value>blue file</Value>
          <Type>Text</Type>
          <Other />
        </RecordsRepositoryProperty>
        <RecordsRepositoryProperty>
          <Name>Name</Name>
          <DisplayName>Name</DisplayName>
          <Value>blue file</Value>
          <Type>Text</Type>
          <Other />
        </RecordsRepositoryProperty>
      </properties>
      <contentTypeName>Document</contentTypeName>
      <originalSaveLocation>http://www.contoso.com/green/Blue-Car.jpg</originalSaveLocation>
    </GetFinalRoutingDestinationFolderUrl>
  </soap:Body>
</soap:Envelope>
```

The protocol server determines the appropriate storage location from the list of configured rules. The protocol server evaluates rules based on the properties of the document if applicable. Once a rule is determined the protocol server returns the data for that final location in the form of the following WSDL message:

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetFinalRoutingDestinationFolderUrlResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/">
      <GetFinalRoutingDestinationFolderUrlResult>
        <Url>http://www.contoso.com/blue</Url>
        <ResultType>ParsingDisabledAtDestination</ResultType>
      </GetFinalRoutingDestinationFolderUrlResult>
    </GetFinalRoutingDestinationFolderUrlResponse>
  </soap:Body>
</soap:Envelope>
```

```

    <CollisionSetting>NoCollision</CollisionSetting>
  </GetFinalRoutingDestinationFolderUrlResult>
</GetFinalRoutingDestinationFolderUrlResponse>
</soap:Body>
</soap:Envelope>

```

If the protocol server determines the final location as "http://www.contoso.com/blue" as in previous example but finds that the file "Blue-Car.jpg" already exists at the location then the protocol server might respond with the following WSDL message:

```

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetFinalRoutingDestinationFolderUrlResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/">
      <GetFinalRoutingDestinationFolderUrlResult>
        <Url>http://www.contoso.com/blue</Url>
        <ResultType>Success</ResultType>
        <CollisionSetting>AppendUniqueSuffixes</CollisionSetting>
      </GetFinalRoutingDestinationFolderUrlResult>
    </GetFinalRoutingDestinationFolderUrlResponse>
  </soap:Body>
</soap:Envelope>

```

On receiving this message the protocol client might decide to append a suffix, such as a unique identifier, before placing the file at the determined location. In this case protocol client might save the file as Blue_123456.jpg at "http://www.contoso.com/blue".

Next, the protocol client might construct the following WSDL message to submit a file named "small.txt" to the repository as a generic "Document" file type with various properties:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SubmitFile xmlns="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/">
      <fileToSubmit>WlpaWlpaWlpaWg</fileToSubmit>
      <properties>
        <RecordsRepositoryProperty>
          <Name>ID</Name>
          <Value>7</Value>
          <Type>Counter</Type>
          <Other/>
        </RecordsRepositoryProperty>
        <RecordsRepositoryProperty>
          <Name>ContentTypeId</Name>
          <Value>0x01010093344DD323F47F48A62CD1F5C002446B</Value>
          <Type>ContentTypeId</Type>
          <Other/>
        </RecordsRepositoryProperty>
        <RecordsRepositoryProperty>
          <Name>ContentType</Name>
          <Value>Document</Value>
        </RecordsRepositoryProperty>
      </properties>
    </SubmitFile>
  </soap:Body>
</soap:Envelope>

```

```

        <Type>Text</Type>
        <Other/>
    </RecordsRepositoryProperty>
    <RecordsRepositoryProperty>
        <Name>Created</Name>
        <Value>1/16/2008 6:18:53 PM</Value>
        <Type>DateTime</Type>
        <Other/>
    </RecordsRepositoryProperty>
    <RecordsRepositoryProperty>
        <Name>vti_modifiedby</Name>
        <Value>CONTOSO\someone</Value>
        <Type>String</Type>
        <Other/>
    </RecordsRepositoryProperty>
</properties>
<recordRouting>Document</recordRouting>
<sourceUrl>http://www.contoso.com/Shared Documents/small.txt</sourceUrl>
    <userName>CONTOSO\someone</userName>
</SubmitFile>
</soap:Body>
</soap:Envelope>

```

The protocol server determines the location in the repository to store the file using the rule for the "Document" file type and successfully stores the file there. The protocol server then responds with the following WSDL message:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body><SubmitFileResponse
    xmlns="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/">
    <SubmitFileResult>&lt;ResultCode&gt;Success&lt;/ResultCode&gt; &lt;AdditionalInformation&gt;ht
    tp://www.contoso.com/sites/RC/MyDocuments/2008-01-17T08-50-
    50Z/small_T2CO2V.txt&lt;/AdditionalInformation&gt;</SubmitFileResult>
    </SubmitFileResponse>
  </soap:Body>
</soap:Envelope>

```

A protocol client might construct the following WSDL message to retrieve information about all the legal holds in the repository.

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetHoldsInfo xmlns="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/"
  />
  </soap:Body>
</soap:Envelope>

```

The protocol server returns the data for all legal holds in the repository in the form of the following WSDL message:

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetHoldsInfoResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/">
      <GetHoldsInfoResult>
        <HoldInfo>
          <Url>http://www.contoso.com/Lists/Holds/DispForm.aspx?ID=1</Url>
          <Title>Contoso vs. SomeCompany</Title>
          <Description>Court case of Contoso vs. SomeCompany</Description>
          <ManagedBy>Joe</ManagedBy>
          <RepositoryName>Records</RepositoryName>
          <Id>1</Id>
          <ListId>8f5f2a74-4fa6-4072-8227-328ff2b01efd</ListId>
          <WebId>9f0ccd71-0b29-4eb3-b5d1-fe4c58f37f21</WebId>
        </HoldInfo>
        <HoldInfo>
          <Url>http://www.contoso.com/Lists/Holds/DispForm.aspx?ID=2</Url>
          <Title>Contoso litigation</Title>
          <Description>Files on hold for the investigation of the "Contoso
litigation"</Description>
          <ManagedBy>Joe</ManagedBy>
          <RepositoryName>Records</RepositoryName>
          <Id>2</Id>
          <ListId>8f5f2a74-4fa6-4072-8227-328ff2b01efd</ListId>
          <WebId>9f0ccd71-0b29-4eb3-b5d1-fe4c58f37f21</WebId>
        </HoldInfo>
      </GetHoldsInfoResult>
    </GetHoldsInfoResponse>
  </soap:Body>
</soap:Envelope>
```

A protocol client might construct the following WSDL message to submit a file to a repository and place it on a legal hold.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SubmitFile xmlns="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/">
      <fileToSubmit> WlpaWlpaWlpaWg </fileToSubmit>
      <properties>
        <RecordsRepositoryProperty>
          <Name>Color</Name>
          <DisplayName>Color</DisplayName>
          <Value>blue</Value>
          <Type>Text</Type>
          <Other />
        </RecordsRepositoryProperty>
        <RecordsRepositoryProperty>
          <Name>Title</Name>
          <DisplayName>Title</DisplayName>
```

```

    <Value>blue file</Value>
    <Type>Text</Type>
    <Other />
  </RecordsRepositoryProperty>
  <RecordsRepositoryProperty>
    <Name>Name</Name>
    <DisplayName>Name</DisplayName>
    <Value>blue file</Value>
    <Type>Text</Type>
    <Other />
  </RecordsRepositoryProperty>
  <RecordsRepositoryProperty>
    <Name>_dlc_hold_url</Name>
    <DisplayName />
    <Value>http://www.contoso.com/Lists/Holds/DispForm.aspx?ID=1</Value>
    <Type>OfficialFileCustomType</Type>
    <Other />
  </RecordsRepositoryProperty>
  <RecordsRepositoryProperty>
    <Name>_dlc_hold_searchquery</Name>
    <DisplayName />
    <Value>contoso</Value>
    <Type>OfficialFileCustomType</Type>
    <Other />
  </RecordsRepositoryProperty>
  <RecordsRepositoryProperty>
    <Name>_dlc_hold_searchcontexturl</Name>
    <DisplayName />
    <Value>http://www.billing.contoso.com</Value>
    <Type>OfficialFileCustomType</Type>
    <Other />
  </RecordsRepositoryProperty>
  <RecordsRepositoryProperty>
    <Name>_dlc_hold_id</Name>
    <DisplayName />
    <Value>2</Value>
    <Type>OfficialFileCustomType</Type>
    <Other />
  </RecordsRepositoryProperty>
  <RecordsRepositoryProperty>
    <Name>_dlc_hold_comments</Name>
    <DisplayName />
    <Value>contoso case search query</Value>
    <Type>OfficialFileCustomType</Type>
    <Other />
  </RecordsRepositoryProperty>
</properties>
<recordRouting>Document</recordRouting>
<sourceUrl>http://www.contoso.com/bills/blue-2006.doc</sourceUrl>
<userName>redmond\pkmacct</userName>
</SubmitFile>
</soap:Body>
</soap:Envelope>

```

The protocol server returns the result of the operation in the form of the following WSDL message:

```
<?xml version="1.0" encoding="utf-8"?>
```



```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SubmitFileResponse
xmlns="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/">
      <SubmitFileResult>&lt;ResultCode&gt;Success&lt;/ResultCode&gt;&lt;ResultUrl&gt;http://co745-
15/blue/blue-
2006.doc&lt;/ResultUrl&gt;&lt;CustomProcessingResult&gt;&lt;HoldsProcessingResult&gt;Success&
lt;/HoldsProcessingResult&gt;&lt;/CustomProcessingResult&gt;</SubmitFileResult>
    </SubmitFileResponse>
  </soap:Body>
</soap:Envelope>
```

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 full WSDL is provided in this appendix.

```
<?xml version="1.0"?>
<wSDL:definitions xmlns:tns="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s1="http://microsoft.com/wsdl/types/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/">
  <wSDL:types>
    <xs:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/"
<xs:import namespace="http://microsoft.com/wsdl/types/">
    <xs:element name="GetHoldsInfo">
      <xs:complexType/>
    </xs:element>
    <xs:element name="GetHoldsInfoResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="GetHoldsInfoResult"
type="tns:ArrayOfHoldInfo"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:complexType name="ArrayOfHoldInfo">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="HoldInfo" nillable="true"
type="tns:HoldInfo"/>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="HoldInfo">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Url" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Title" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Description" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ManagedBy" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="RepositoryName" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="DiscoveryQueries" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="Id" type="xs:int"/>
        <xs:element minOccurs="1" maxOccurs="1" name="ListId" type="s1:guid"/>
        <xs:element minOccurs="1" maxOccurs="1" name="WebId" type="s1:guid"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="GetFinalRoutingDestinationFolderUrl">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="properties"
type="tns:ArrayOfRecordsRepositoryProperty"/>
          <xs:element minOccurs="0" maxOccurs="1" name="contentTypeName" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="originalSaveLocation"
type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:complexType name="ArrayOfRecordsRepositoryProperty">
```

```

    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="RecordsRepositoryProperty"
nillable="true" type="tns:RecordsRepositoryProperty"/>
    </xs:sequence>
  </xs:complexType>
<xs:complexType name="RecordsRepositoryProperty">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="DisplayName" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Value" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Type" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Other" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="GetFinalRoutingDestinationFolderUrlResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1"
name="GetFinalRoutingDestinationFolderUrlResult" type="tns:DocumentRoutingResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="DocumentRoutingResult">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Url" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="ResultType"
type="tns:DocumentRoutingResultType"/>
    <xs:element minOccurs="1" maxOccurs="1" name="CollisionSetting"
type="tns:DocumentRoutingCollisionSetting"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="DocumentRoutingResultType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Success"/>
    <xs:enumeration value="SuccessToDropOffLibrary"/>
    <xs:enumeration value="MissingRequiredProperties"/>
    <xs:enumeration value="NoMatchingRules"/>
    <xs:enumeration value="DocumentRoutingDisabled"/>
    <xs:enumeration value="PermissionDeniedAtDestination"/>
    <xs:enumeration value="ParsingDisabledAtDestination"/>
    <xs:enumeration value="OriginalSaveLocationIsDocumentSet"/>
    <xs:enumeration value="NoEnforcementAtOriginalSaveLocation"/>
    <xs:enumeration value="UnknownFailure"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="DocumentRoutingCollisionSetting">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NoCollision"/>
    <xs:enumeration value="UseSharePointVersioning"/>
    <xs:enumeration value="AppendUniqueSuffixes"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="SubmitFile">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="fileToSubmit"
type="xs:base64Binary"/>
      <xs:element minOccurs="0" maxOccurs="1" name="properties"
type="tns:ArrayOfRecordsRepositoryProperty"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        <xs:element minOccurs="0" maxOccurs="1" name="recordRouting" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="sourceUrl" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="userName" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="SubmitFileResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="SubmitFileResult"
type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetServerInfo">
    <xs:complexType/>
</xs:element>
<xs:element name="GetServerInfoResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetServerInfoResult"
type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetRecordRoutingCollection">
    <xs:complexType/>
</xs:element>
<xs:element name="GetRecordRoutingCollectionResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetRecordRoutingCollectionResult"
type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetRecordRouting">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="recordRouting" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetRecordRoutingResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetRecordRoutingResult"
type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="CustomProcessingResult">
    <xs:sequence>
        <xs:element minOccurs="0" name="HoldsProcessingResult"
type="tns:HoldProcessingResult"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="SubmitFileResult">
    <xs:sequence>

```

```

    <xs:element name="ResultCode" type="tns:SubmitFileResultCode"/>
    <xs:choice>
      <xs:element minOccurs="0" name="ResultUrl" type="xs:anyURI"/>
      <xs:element minOccurs="0" name="AdditionalInformation" type="xs:string"/>
    </xs:choice>
    <xs:element minOccurs="0" name="CustomProcessingResult"
type="tns:CustomProcessingResult"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ServerInfo">
  <xs:sequence>
    <xs:element name="ServerType" type="xs:string"/>
    <xs:element name="ServerVersion" type="xs:string"/>
    <xs:element name="RoutingWeb" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="SubmitFileResultCode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Success"/>
    <xs:enumeration value="MoreInformation"/>
    <xs:enumeration value="InvalidRouterConfiguration"/>
    <xs:enumeration value="InvalidArgument"/>
    <xs:enumeration value="InvalidUser"/>
    <xs:enumeration value="NotFound"/>
    <xs:enumeration value="FileRejected"/>
    <xs:enumeration value="UnknownError"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="HoldProcessingResult">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Success"/>
    <xs:enumeration value="Failure"/>
    <xs:enumeration value="InDropOffZone"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
<xs:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/wsdl/types/">
  <xs:simpleType name="guid">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-
9a-fA-F]{12}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
</wsdl:types>
<wsdl:message name="GetHoldsInfoSoapIn">
  <wsdl:part name="parameters" element="tns:GetHoldsInfo"/>
</wsdl:message>
<wsdl:message name="GetHoldsInfoSoapOut">
  <wsdl:part name="parameters" element="tns:GetHoldsInfoResponse"/>
</wsdl:message>
<wsdl:message name="GetFinalRoutingDestinationFolderUrlSoapIn">
  <wsdl:part name="parameters" element="tns:GetFinalRoutingDestinationFolderUrl"/>
</wsdl:message>
<wsdl:message name="GetFinalRoutingDestinationFolderUrlSoapOut">
  <wsdl:part name="parameters" element="tns:GetFinalRoutingDestinationFolderUrlResponse"/>
</wsdl:message>
<wsdl:message name="SubmitFileSoapIn">

```

```

    <wsdl:part name="parameters" element="tns:SubmitFile"/>
  </wsdl:message>
  <wsdl:message name="SubmitFileSoapOut">
    <wsdl:part name="parameters" element="tns:SubmitFileResponse"/>
  </wsdl:message>
  <wsdl:message name="GetServerInfoSoapIn">
    <wsdl:part name="parameters" element="tns:GetServerInfo"/>
  </wsdl:message>
  <wsdl:message name="GetServerInfoSoapOut">
    <wsdl:part name="parameters" element="tns:GetServerInfoResponse"/>
  </wsdl:message>
  <wsdl:message name="GetRecordRoutingCollectionSoapIn">
    <wsdl:part name="parameters" element="tns:GetRecordRoutingCollection"/>
  </wsdl:message>
  <wsdl:message name="GetRecordRoutingCollectionSoapOut">
    <wsdl:part name="parameters" element="tns:GetRecordRoutingCollectionResponse"/>
  </wsdl:message>
  <wsdl:message name="GetRecordRoutingSoapIn">
    <wsdl:part name="parameters" element="tns:GetRecordRouting"/>
  </wsdl:message>
  <wsdl:message name="GetRecordRoutingSoapOut">
    <wsdl:part name="parameters" element="tns:GetRecordRoutingResponse"/>
  </wsdl:message>
  <wsdl:portType name="OfficialFileSoap">
    <wsdl:operation name="GetHoldsInfo">
      <wsdl:input message="tns:GetHoldsInfoSoapIn"/>
      <wsdl:output message="tns:GetHoldsInfoSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="GetFinalRoutingDestinationFolderUrl">
      <wsdl:input message="tns:GetFinalRoutingDestinationFolderUrlSoapIn"/>
      <wsdl:output message="tns:GetFinalRoutingDestinationFolderUrlSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="SubmitFile">
      <wsdl:input message="tns:SubmitFileSoapIn"/>
      <wsdl:output message="tns:SubmitFileSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="GetServerInfo">
      <wsdl:input message="tns:GetServerInfoSoapIn"/>
      <wsdl:output message="tns:GetServerInfoSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="GetRecordRoutingCollection">
      <wsdl:input message="tns:GetRecordRoutingCollectionSoapIn"/>
      <wsdl:output message="tns:GetRecordRoutingCollectionSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="GetRecordRouting">
      <wsdl:input message="tns:GetRecordRoutingSoapIn"/>
      <wsdl:output message="tns:GetRecordRoutingSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="OfficialFileSoap" type="tns:OfficialFileSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetHoldsInfo">
      <soap:operation
        soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetHoldsInfo"
        style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>

```

```

        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetFinalRoutingDestinationFolderUrl">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetFinalRoutingDes
tinationFolderUrl" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SubmitFile">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/SubmitFile"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetServerInfo">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetServerInfo"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetRecordRoutingCollection">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetRecordRoutingCo
llection" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetRecordRouting">
    <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetRecordRouting"
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="OfficialFileSoap12" type="tns:OfficialFileSoap">

```



```

    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetHoldsInfo">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetHoldsInfo"
style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetFinalRoutingDestinationFolderUrl">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetFinalRoutingDes
tinationFolderUrl" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SubmitFile">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/SubmitFile"
style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetServerInfo">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetServerInfo"
style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetRecordRoutingCollection">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetRecordRoutingCo
llection" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetRecordRouting">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/recordsrepository/GetRecordRouting"
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: Full XML Schema

8 Appendix C: Product Behavior

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

- Microsoft® Office SharePoint® Server 2007
- Microsoft® SharePoint® Server 2010
- Microsoft® SharePoint® Server 2013

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:](#) SharePoint Server 2010 and SharePoint Server 2013 use the URL of the **site** where the search query was performed.

[<2> Section 2.2.4.3:](#) Office SharePoint Server 2007 does not use this element. SharePoint Server 2010 and SharePoint Server 2013 do not specify this element if the property is not a **field** of the file being submitted.

[<3> Section 2.2.4.3:](#) Office SharePoint Server 2007 does not use this format.

[<4> Section 2.2.4.4:](#) Office SharePoint Server 2007 does not use this element.

[<5> Section 3.1.4:](#) Office SharePoint Server 2007 does not provide this method.

[<6> Section 3.1.4:](#) Office SharePoint Server 2007 does not provide this method.

[<7> Section 3.1.4.1:](#) Office SharePoint Server 2007 does not provide this method.

[<8> Section 3.1.4.2:](#) Office SharePoint Server 2007 does not provide this method.

[<9> Section 3.1.4.5:](#) Office SharePoint Server 2007 returns "Microsoft.Office.Server" as the ServerType and "Microsoft.Office.OfficialFileSoap, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" as the ServerVersion. SharePoint Server 2010 returns "Microsoft.Office.Server v4" as the ServerType and " Microsoft.Office.OfficialFileSoap, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" as the ServerVersion. SharePoint Server 2013 returns "Microsoft.Office.Server v4" as the ServerType and " Microsoft.Office.OfficialFileSoap, Version=15.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" as the ServerVersion.

[<10> Section 3.1.4.6:](#) Third-party modules can be added to Office SharePoint Server 2007, SharePoint Server 2010, and SharePoint Server 2013 which can cause the submission to be rejected.

[<11> Section 3.1.4.6:](#) Office SharePoint Server 2007 does not include the **CustomProcessingResult** element. The **CustomProcessingResult** element is not set if the properties listed in step 12 are not present.

<12> [Section 3.1.4.6:](#) Office SharePoint Server 2007 does not include the **CustomProcessingResult** element.

<13> [Section 3.1.4.6:](#) Office SharePoint Server 2007 returns the URL in the **AdditionalInformation** element.

<14> [Section 3.1.4.6:](#) Office SharePoint Server 2007 does not include the **CustomProcessingResult** element.

<15> [Section 3.1.4.6:](#) Office SharePoint Server 2007 does not include the **CustomProcessingResult** element.

<16> [Section 3.1.4.6.2.1:](#) The behavior is undefined if the **userName** element is SHAREPOINT\system for Office SharePoint Server 2007, SharePoint Server 2010, and SharePoint Server 2013.

9 Change Tracking

This section identifies changes that were made to the [MS-OFFICIALFILE] protocol document between the September 2012 and October 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

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

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

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

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

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

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.3 Overview	Changed the name from 'Protocol Overview (Synopsis)' to 'Overview'.	N	Content updated for template compliance.
2.2.4.5 SubmitFileResult	Added section to contain content moved from the Complex Types section in the SubmitFile section.	N	Content updated.
3.1.4.6.2.2 SubmitFileResponse	Update the schema to be consistent with the product wsdl by changing type to string.	N	Content updated.
3.1.4.6.3 Complex Types	Moved SubmitFileResult information to the SubmitFileResult section.	N	Content updated.

10 Index

A

Abstract data model
 [server](#) 16
[Applicability](#) 9
[ArrayOfRecordsRepositoryProperty complex type](#) 11
[Attribute groups](#) 15
[Attributes](#) 15

C

[Capability negotiation](#) 9
[Change tracking](#) 54
Common data structures ([section 2.2.9](#) 15, [section 2.2.9](#) 15)
[Complex types](#) 11
 [ArrayOfRecordsRepositoryProperty](#) 11
 [CustomProcessingResult](#) 11
 [RecordsRepositoryProperty](#) 11
 [ServerInfo](#) 12
 [SubmitFileResult](#) 13
[CustomProcessingResult complex type](#) 11

D

Data model - abstract
 [server](#) 16

E

Events
 [local - server](#) 35
 [timer - server](#) 35
Examples
 [overview](#) 36

F

[Fields - vendor-extensible](#) 9
[Full WSDL](#) 43

G

[Glossary](#) 7
[Groups](#) 15

H

[HoldProcessingResult simple type](#) 14

I

[Implementer - security considerations](#) 42
[Index of security parameters](#) 42
[Informative references](#) 8
Initialization
 [server](#) 17
[Introduction](#) 7

L

Local events
 [server](#) 35

M

Message processing
 [server](#) 17
Messages
 [ArrayOfRecordsRepositoryProperty complex type](#) 11
 [attribute groups](#) 15
 [attributes](#) 15
 common data structures ([section 2.2.9](#) 15, [section 2.2.9](#) 15)
 [complex types](#) 11
 [CustomProcessingResult complex type](#) 11
 [elements](#) 11
 [enumerated](#) 10
 [groups](#) 15
 [HoldProcessingResult simple type](#) 14
 [namespaces](#) 10
 [RecordsRepositoryProperty complex type](#) 11
 [ServerInfo complex type](#) 12
 [simple types](#) 13
 [SubmitFileResult complex type](#) 13
 [SubmitFileResultCode simple type](#) 14
 [syntax](#) 10
 [transport](#) 10

N

[Namespaces](#) 10
[Normative references](#) 8

O

Operations
 [GetFinalRoutingDestinationFolderUrl](#) 17
 [GetHoldsInfo](#) 22
 [GetRecordRouting](#) 26
 [GetRecordRoutingCollection](#) 27
 [GetServerInfo](#) 29
 [SubmitFile](#) 31
[Overview \(synopsis\)](#) 8

P

[Parameters - security index](#) 42
[Preconditions](#) 9
[Prerequisites](#) 9
[Product behavior](#) 52

R

[RecordsRepositoryProperty complex type](#) 11

[References](#) 7
 [informative](#) 8
 [normative](#) 8
[Relationship to other protocols](#) 9

S

Security
 [implementer considerations](#) 42
 [parameter index](#) 42
Sequencing rules
 [server](#) 17
Server
 [abstract data model](#) 16
 [GetFinalRoutingDestinationFolderUrl operation](#) 17
 [GetHoldsInfo operation](#) 22
 [GetRecordRouting operation](#) 26
 [GetRecordRoutingCollection operation](#) 27
 [GetServerInfo operation](#) 29
 [initialization](#) 17
 [local events](#) 35
 [message processing](#) 17
 [sequencing rules](#) 17
 [SubmitFile operation](#) 31
 [timer events](#) 35
 [timers](#) 16
 [ServerInfo complex type](#) 12
 [Simple types](#) 13
 [HoldProcessingResult](#) 14
 [SubmitFileResultCode](#) 14
 [Standards assignments](#) 9
 [SubmitFileResult complex type](#) 13
 [SubmitFileResultCode simple type](#) 14
Syntax
 [messages - overview](#) 10

T

Timer events
 [server](#) 35
Timers
 [server](#) 16
[Tracking changes](#) 54
[Transport](#) 10
Types
 [complex](#) 11
 [simple](#) 13

V

[Vendor-extensible fields](#) 9
[Versioning](#) 9

W

[WSDL](#) 43