[MS-OXWSSRCH]:
Mailbox Search 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.

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final

documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
07/15/2009	1.0	Major	Initial Availability.
11/04/2009	1.1.0	Minor	Updated the technical content.
02/10/2010	1.1.0	None	Version 1.1.0 release
05/05/2010	1.1.1	Editorial	Revised and edited the technical content.
08/04/2010	2.0	Major	Significantly changed the technical content.
11/03/2010	2.1	Minor	Clarified the meaning of the technical content.
03/18/2011	3.0	Major	Significantly changed the technical content.
08/05/2011	4.0	Major	Significantly changed the technical content.
10/07/2011	5.0	Major	Significantly changed the technical content.
01/20/2012	6.0	Major	Significantly changed the technical content.
04/27/2012	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	7.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction 6
1.1 Glossary 6
1.2 References 6
1.2.1 Normative References 6
1.2.2 Informative References 7
1.3 Overview 8
1.4 Relationship to Other Protocols 8
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 10
2.2.3.1 tns:And Element 11
2.2.3.2 tns:Contains Element 12
2.2.3.3 tns:Excludes Element 12
2.2.3.4 tns:Exists Element 12
2.2.3.5 tns:IsEqualTo Element 12
2.2.3.6 tns:IsGreaterThan Element 12
2.2.3.7 tns:IsGreaterThanOrEqualTo Element 13
2.2.3.8 tns:IsLessThan Element 13
2.2.3.9 tns:IsLessThanOrEqualTo Element 13
2.2.3.10 tns:IsNotEqualTo Element 13
2.2.3.11 tns:Not Element 13
2.2.3.12 tns:Or Element 14
2.2.3.13 tns:SearchExpression Element 14
2.2.4 Complex Types 14
2.2.4.1 m:FindFolderResponseMessageType Complex Type 16
2.2.4.2 m:FindItemResponseMessageType Complex Type 16
2.2.4.3 t:AndType Complex Type 17
2.2.4.4 t:ArrayOfGroupedItemsType Complex Type 17
2.2.4.5 t:BasePagingType Complex Type 18
2.2.4.6 t:ContainsExpressionType Complex Type 18
2.2.4.7 t:ExcludesType Complex Type 19
2.2.4.8 t:ExcludesValueType Complex Type 20
2.2.4.9 t:ExistsType Complex Type 20
2.2.4.10 t:FindFolderParentType Complex Type 21
2.2.4.11 t:FindItemParentType Complex Type 21
2.2.4.12 t:FractionalPageViewType Complex Type 22
2.2.4.13 t:GroupedItemsType Complex Type 23
2.2.4.14 t:IndexedPageViewType Complex Type 23
2.2.4.15 t:IsEqualToType Complex Type 24
2.2.4.16 t:IsGreaterThanOrEqualToType Complex Type 24
2.2.4.17 t:IsGreaterThanType Complex Type 25

2.2.4.18 t:IsLessThanOrEqualToType Complex Type 25
2.2.4.19 t:IsLessThanType Complex Type 25
2.2.4.20 t:IsNotEqualToType Complex Type 26
2.2.4.21 t:MultipleOperandBooleanExpressionType Complex Type 26
2.2.4.22 t:NotType Complex Type 27
2.2.4.23 t:OrType Complex Type 27
2.2.4.24 t:RestrictionType Complex Type 27
2.2.4.25 t:SearchExpressionType Complex Type 28
2.2.4.26 t:SearchFolderType Complex Type 28
2.2.4.27 t:SearchParametersType Complex Type 28
2.2.4.28 t:SeekToConditionPageViewType Complex Type 29
2.2.4.29 t:TwoOperandExpressionType Complex Type 30
2.2.5 Simple Types 31
2.2.5.1 t:ContainmentComparisonType Simple Type 31
2.2.5.2 t:ContainmentModeType Simple Type 32
2.2.5.3 t:FolderQueryTraversalType Simple Type 33
2.2.5.4 t:IndexBasePointType Simple Type 34
2.2.5.5 t:ItemQueryTraversalType Simple Type 34
2.2.5.6 t:SearchFolderTraversalType Simple Type 35
2.2.6 Attributes 35
2.2.7 Groups 35
2.2.8 Attribute Groups 35

3 Protocol Details 36
3.1 ExchangeServicePortType Server Details 36
3.1.1 Abstract Data Model 36
3.1.2 Timers 36
3.1.3 Initialization 36
3.1.4 Message Processing Events and Sequencing Rules 36
3.1.4.1 FindFolder Operation 36
3.1.4.1.1 Messages 37
3.1.4.1.1.1 tns:FindFolderSoapIn Message 37
3.1.4.1.1.2 tns:FindFolderSoapOut Message 38
3.1.4.1.2 Elements 38
3.1.4.1.2.1 tns:FindFolder Element 39
3.1.4.1.2.2 tns:FindFolderResponse Element 39
3.1.4.1.3 Complex Types 39
3.1.4.1.3.1 m:FindFolderResponseType Complex Type 39
3.1.4.1.3.2 m:FindFolderType Complex Type 39
3.1.4.2 FindItem Operation 41
3.1.4.2.1 Messages 41
3.1.4.2.1.1 tns:FindItemSoapIn Message 42
3.1.4.2.1.2 tns:FindItemSoapOut Message 43
3.1.4.2.2 Elements 43
3.1.4.2.2.1 tns:FindItem Element 43
3.1.4.2.2.2 tns:FindItemResponse Element 43
3.1.4.2.3 Complex Types 44
3.1.4.2.3.1 m:FindItemResponseType Complex Type 44
3.1.4.2.3.2 m:FindItemType Complex Type 44
3.1.4.2.3.3 t:AggregateOnType Complex Type 48
3.1.4.2.3.4 t:BaseGroupByType Complex Type 49
3.1.4.2.3.5 t:DistinguishedGroupByType Complex Type 49
3.1.4.2.3.6 t:FieldOrderType Complex Type 50

3.1.4.2.3.7 t:GroupByType Complex Type 50
3.1.4.2.3.8 t:NonEmptyArrayOfFieldOrdersType Complex Type 51
3.1.4.2.4 Simple Types 51
3.1.4.2.4.1 t:AggregateType Simple Type 52
3.1.4.2.4.2 t:SortDirectionType Simple Type 52
3.1.4.2.4.3 t:StandardGroupByType Simple Type 53
3.1.5 Timer Events 53
3.1.6 Other Local Events 53

4 Protocol Examples 54

5 Security 55
5.1 Security Considerations for Implementers 55
5.2 Index of Security Parameters 55

6 Appendix A: Full WSDL 56

7 Appendix B: Full XML Schema 59
7.1 Messages Schema 59
7.2 Types Schema 61

8 Appendix C: Product Behavior 67

9 Change Tracking 68

10 Index 70

1 Introduction

The Mailbox Search Web Service Protocol is used to search the contents of a server and return items and groups of items.

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]:

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

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

Deleted Items folder
endpoint
mailbox
search folder
Simple Object Access Protocol (SOAP)
SOAP action
SOAP body
SOAP header
SOAP message
store
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
Web Services Description Language (WSDL)
WSDL message
WSDL port type
XML namespace
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-OXWSCDATA] Microsoft Corporation, "Common Web Service Data Types".

[MS-OXWSCONT] Microsoft Corporation, "Contacts Web Service Protocol Specification".

[MS-OXWSDLIST] Microsoft Corporation, "Distribution List Creation and Usage Web Service Protocol Specification".

[MS-OXWSFOLD] Microsoft Corporation, "Folders and Folder Permissions Web Service Protocol Specification".

[MS-OXWSGTZ] Microsoft Corporation, "Get Server Time Zone Web Service Protocol Specification".

[MS-OXWSMSG] Microsoft Corporation, "E-Mail Message Types Web Service Protocol Specification".

[MS-OXWSMTGS] Microsoft Corporation, "Calendaring Web Service Protocol Specification".

[MS-OXWSPOST] Microsoft Corporation, "Post Items Web Service Protocol Specification".

[MS-OXWSTASK] Microsoft Corporation, "Tasks Web Service Protocol Specification".

[MS-OXWSXPROP] Microsoft Corporation, "Extended Properties Structure".

[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

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

[RFC3066] Alvestrand, H., "Tags for the Identification of Language", RFC 3066, January 2001, http://www.ietf.org/rfc/rfc3066.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/

[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/

[XMLSCHEMA0] Fallside, D., Ed. and Walmsley, P., Ed., "XML Schema Part 0: Primer, Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/

[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-OXDSCLI] Microsoft Corporation, "Autodiscover Publishing and Lookup Protocol Specification".

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary".

[MS-OXWSADISC] Microsoft Corporation, "Autodiscover Publishing and Lookup SOAP-Based Web Service Protocol Specification".

[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

1.3 Overview

This protocol provides clients with operations that enable them to search the contents of a server data **store** and to return the results of that search.

1.4 Relationship to Other Protocols

A client that implements this protocol can use the Autodiscover Publishing and Lookup SOAP-based Web Service Protocol, as described in [MS-OXWSADISC], or the Autodiscover Publishing and Lookup Protocol, as described in [MS-OXDSCLI], to identify the target **endpoint (4)** to use for each operation.

This protocol uses the SOAP Protocol, as described in [SOAP1.1], to describe the structure information that is exchanged between the client and server. This protocol uses the XML Protocol, as described in [XMLSCHEMA1] and [XMLSCHEMA2], to describe the message content that is sent to and from the server.

This protocol uses **SOAP** over **HTTP**, as described in [RFC2616], and SOAP over **HTTPS**, as described in [RFC2818], as shown in the following layering diagram.

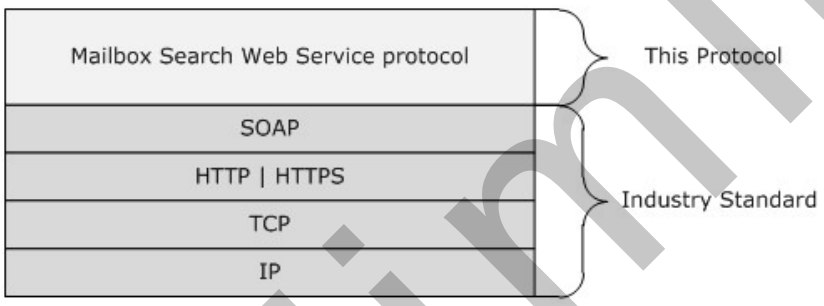

Figure 1: This protocol in relation to other protocols

This protocol specifies searches that identify items in the server store. After the item identifier is returned, one of the following protocols is used to return the information from the store:

- Folders and Folder Permissions Web Service Protocol, as described in [MS-OXWSFOLD]
- E-Mail Message Types Web Service Protocol, as described in [MS-OXWSMSG]
- Calendaring Web Service Protocol, as described in [MS-OXWSMTGS]
- Post Items Web Service Protocol, as described in [MS-OXWSPOST]
- Tasks Web Service Protocol, as described in [MS-OXWSTASK]

1.5 Prerequisites/Preconditions

The endpoint (4) **URL** that is returned by either the Autodiscover Publishing Lookup SOAP-Based Web Service Protocol, as described in [MS-OXWSADISC], or the Autodiscover Publishing Lookup Protocol, as described in [MS-OXDSCLI], is required to form the HTTP request to the Web server that hosts this protocol. The operations that this protocol defines cannot be accessed unless the correct endpoint (4) is identified in the HTTP Web requests that target this protocol.

1.6 Applicability Statement

This protocol is applicable to client applications that search the contents of the server store.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with SOAP 1.1, as specified in section 2.1.
- **Protocol Versions:** This protocol specifies only one **WSDL port type** version. The WSDL version of the request is identified by using the **RequestServerVersion** element, as described in [MS-OXWSCDATA] section 2.2.5.9, and the version of the server responding to the request is identified by using the **ServerVersionInfo** element, as described in [MS-OXWSCDATA] section 2.2.5.10.
- **Security and Authentication Methods:** This protocol relies on the Web server that is hosting it to perform authentication.
- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section 3.1.4.
- **Capability Negotiation:** This protocol does not support version negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The **WSDL** in this specification provides a base description of the protocol. The schema in this specification provides a base description of the message syntax. The text that specifies the WSDL and schema might specify restrictions that reflect actual protocol 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**, or **present**.

2.1 Transport

The SOAP version that is supported is SOAP 1.1. For details, see [SOAP1.1].

This protocol relies on the Weber server that hosts the application to perform authentication. The protocol SHOULD use secure communications via HTTPS, as defined in [RFC2818].

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 defined in [XMLSCHEMA1] and [XMLSCHEMA2], and Web Services Description Language (WSDL), as defined in [WSDL].

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/exchange/services/2006/messages	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
targetNamespace	http://schemas.microsoft.com/exchange/services/2006/messages	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	

2.2.2 Messages

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

2.2.3 Elements

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

Element name	Description
And	Specifies a search expression that allows you to perform a Boolean **AND** operation between two or more search expressions.
Contains	Specifies a search expression that determines whether a given property contains the supplied constant string value.
Excludes	Specifies a search expression that allows you to perform a bitwise mask of the specified property and a supplied value.
Exists	Specifies a search expression that determines whether the specified property exists on an item.
IsEqualTo	Specifies a search expression that compares a property value with either a constant value or another property value and evaluates to true if the two are equal.
IsGreaterThan	Specifies a search expression that compares a property with either a constant value or another property value and evaluates to true if the first property is greater than the second.
IsGreaterThanOrEqualTo	Specifies a search expression that compares a property with either a constant value or another property and evaluates to true if the first property is greater than or equal to the second.
IsLessThan	Specifies a search expression that compares a property to either a constant value or another property and evaluates to true if the first property is less than the second.
IsLessThanOrEqualTo	Specifies a search expression that compares a property to either a constant value or another property and evaluates to true if the first property is less than or equal to the second.
IsNotEqualTo	Specifies a search expression that compares a property value with either a constant value or another property value and evaluates to true if the two are not the same.
Not	Specifies a search expression that negates the Boolean value of a search expression that it contains.
Or	Specifies a search expression that performs a Boolean **OR** operation on two or more search expressions.
SearchExpression	Specifies the base schema type for all search expressions. This type is abstract and will never occur directly within instance documents.

2.2.3.1 tns:And Element

The **And** element specifies a search expression that allows you to perform a Boolean **AND** operation between two or more search expressions. The **AND** operation evaluates to **TRUE** if all the search expressions that are contained within the **And** element are **TRUE**.

```
<xs:element name="And"
    type="t:AndType"
    substitutionGroup="t:SearchExpression"
/>
```

2.2.3.2 tns:Contains Element

The **Contains** element specifies a search expression that determines whether a given property contains the supplied constant string value.

```
<xs:element name="Contains"
  type="t:ContainsExpressionType"
  substitutionGroup="t:SearchExpression"
 />
```

2.2.3.3 tns:Excludes Element

The **Excludes** element specifies a search expression that allows you to perform a bitwise mask of the specified property and a supplied value.

```
<xs:element name="Excludes"
  type="t:ExcludesType"
  substitutionGroup="t:SearchExpression"
 />
```

2.2.3.4 tns:Exists Element

The **Exists** element specifies a search expression that determines whether the specified property exists on an item. The **Exists** element evaluates to true if the specified property exists on the item.

```
<xs:element name="Exists"
  type="t:ExistsType"
  substitutionGroup="t:SearchExpression"
/>
```

2.2.3.5 tns:IsEqualTo Element

The **IsEqualTo** element specifies a search expression that compares a property value with either a constant value or another property value and evaluates to true if they are equal.

```
<xs:element name="IsEqualTo"
  type="t:IsEqualToType"
  substitutionGroup="t:SearchExpression"
/>
```

2.2.3.6 tns:IsGreaterThan Element

The **IsGreaterThan** element specifies a search expression that compares a property with either a constant value or another property and evaluates to true if the first property is greater.

```
<xs:element name="IsGreaterThan"
  type="t:IsGreaterThanType"
  substitutionGroup="t:SearchExpression"
/>
```

2.2.3.7 tns:IsGreaterThanOrEqualTo Element

The **IsGreaterThanOrEqualTo** element specifies a search expression that compares a property with either a constant value or another property and evaluates to true if the first property is greater than or equal to the second.

```
<xs:element name="IsGreaterThanOrEqualTo"
  type="t:IsGreaterThanOrEqualToType"
  substitutionGroup="t:SearchExpression"
/>
```

2.2.3.8 tns:IsLessThan Element

The **IsLessThan** element specifies a search expression that compares a property to either a constant value or another property and evaluates to true if the first property is less than the second.

```
<xs:element name="IsLessThan"
  type="t:IsLessThanType"
  substitutionGroup="t:SearchExpression"
/>
```

2.2.3.9 tns:IsLessThanOrEqualTo Element

The **IsLessThanOrEqualTo** element specifies a search expression that compares a property to either a constant value or another property and evaluates to true if the first property is less than or equal to the second.

```
<xs:element name="IsLessThanOrEqualTo"
  type="t:IsLessThanOrEqualToType"
  substitutionGroup="t:SearchExpression"
/>
```

2.2.3.10 tns:IsNotEqualTo Element

The **IsNotEqualTo** element specifies a search expression that compares a property value with either a constant value or another property value and evaluates to true if they are not the same.

```
<xs:element name="IsNotEqualTo"
  type="t:IsNotEqualToType"
  substitutionGroup="t:SearchExpression"
/>
```

2.2.3.11 tns:Not Element

The **Not** element specifies a search expression that negates the Boolean value of a search expression that it contains.

```
<xs:element name="Not"
  type="t:NotType"
 />
```

2.2.3.12 tns:Or Element

The **Or** element specifies a search expression that performs a logical **OR** on the search expressions that it contains and evaluates to true if any of the search expressions evaluate to true.

```
<xs:element name="Or"
  type="t:OrType"
  substitutionGroup="t:SearchExpression"
/>
```

2.2.3.13 tns:SearchExpression Element

The **SearchExpression** element specifies the base schema type for all search expressions. This type is abstract and will never occur directly within instance documents. This type defines a substitution group.

```
<xs:element name="SearchExpression"
  type="t:SearchExpressionType"
 />
```

2.2.4 Complex Types

Note *Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.*

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

Complex type name	Description
FindFolderResponseMessageType	Specifies the result body for the **FindFolder** operation (section 3.1.4.1).
FindItemResponseMessageType	Specifies the result body from the **FindItem** operation (section 3.1.4.2).
AndType	Specifies a search expression that performs a Boolean **AND** operation between two or more search expressions.
ArrayOfGroupedItemsType	Specifies an array of items that are returned by the **FindItem** operation.
BasePagingType	Specifies the base type for derived types that specify paged views.
ContainsExpressionType	Specifies a search expression that determines whether a given property contains the supplied constant string value.
ExcludesType	Specifies a bitwise mask of a property for an exclude

Complex type name	Description
	search restriction.
ExcludesValueType	Specifies a hexadecimal or decimal mask for an exclude search restriction.
ExistsType	Specifies a search restriction that resolves to true if the supplied property exists on an item.
FindFolderParentType	Specifies the results of searching a single root folder.
FindItemParentType	Specifies the results of searching a single root folder.
FractionalPageViewType	Specifies where a paged view starts and the maximum number of items that are returned.
GroupedItemsType	Specifies a collection of items that are the result of a grouped **FindItem** operation.
IndexedPageViewType	Specifies how paged item information is returned by the **FindItem** or the **FindFolder** operation.
IsEqualToType	Specifies a search expression that compares a property with either a constant value or another property and evaluates to "true" if they are equal.
IsGreaterThanOrEqualToType	Specifies a search expression that compares a property with either a constant value or another property and evaluates to "true" if the first property is greater than or equal to the second.
IsGreaterThanType	Specifies a search expression that compares a property with either a constant value or another property and returns "true" if the first property is greater.
IsLessThanOrEqualToType	Specifies a search expression that compares a property with either a constant value or another property and evaluates to true if the first property is less than or equal to the second.
IsLessThanType	Specifies a search expression that compares a property with either a constant value or another property and evaluates to true if the first property is less than the second.
IsNotEqualToType	Specifies a search expression that compares a property with either a constant value or another property and evaluates to true if the values are not the same.
MultipleOperandBooleanExpressionType	Specifies the base type for search expressions that are formed by two or more Boolean operands.
NotType	Specifies a search expression that negates the Boolean value of the search expression that it contains.
OrType	Specifies a search expression that performs a logical **OR** on the search expression that it contains.
RestrictionType	Specifies a search restriction or query for the **FindItem** operation or the **FindFolder** operation.

Complex type name	Description
SearchExpressionType	Specifies the base type for all search expressions.
SearchFolderType	Specifies a representation of a **search folder** that is contained in a **mailbox**.
SearchParametersType	Specifies the search parameters that define the contents of a search folder.
SeekToConditionPageViewType	Specifies a condition that has to be met in order for an item to be included in the result set that is returned by the **FindItem** operation or the **FindFolder** operation.<1>
TwoOperandExpressionType	Specifies the base type for derived classes that represent a restriction that is formed by comparing two values against one another.

2.2.4.1 m:FindFolderResponseMessageType Complex Type

The **FindFolderResponseMessageType** complex type specifies the response message for the **FindFolder** operation, as specified in section 3.1.4.1. The **FindFolderResponseMessageType** complex type extends the **ResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.57.

```
<xs:complexType name="FindFolderResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="RootFolder"
          type="t:FindFolderParentType"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **FindFolderResponseMessageType** complex type.

Element name	Type	Description
RootFolder	**t:FindFolderParentType** (section 2.2.4.10)	Specifies an array of folders and paging information that is returned by the **FindFolder** operation. .

2.2.4.2 m:FindItemResponseMessageType Complex Type

The **FindItemResponseMessageType** complex type specifies the result body from the **FindItem** operation, as specified in section 3.1.4.2. The **FindItemResponseMessageType** complex type extends the **ResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.57.

```
<xs:complexType name="FindItemResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="RootFolder"
          type="t:FindFolderParentType"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **FindItemResponseMessageType** complex type.

Element name	Type	Description
RootFolder	**t:FindFolderParentType** (section 2.2.4.10)	Specifies the root folder of the reponse.

2.2.4.3 t:AndType Complex Type

The **AndType** complex type specifies a search expression that performs a Boolean **AND** operation between two or more search expressions. The result of the **AND** operation is **TRUE** if all the search expressions that are contained within the **AndType** complex type are true. The **AndType** complex type extends the **MultipleOperandBooleanExpressionType** complex type, as specified in section 2.2.4.21.

```
<xs:complexType name="AndType">
  <xs:complexContent>
    <xs:extension
      base="t:MultipleOperandBooleanExpressionType"
    />
  </xs:complexContent>
</xs:complexType>
```

2.2.4.4 t:ArrayOfGroupedItemsType Complex Type

The **ArrayOfGroupedItemsType** complex type specifies an array of items that are returned by the **FindItem** operation, as specified in section 3.1.4.2.

```
<xs:complexType name="ArrayOfGroupedItemsType">
  <xs:choice>
    <xs:element name="GroupedItems"
      type="t:GroupedItemsType"
    />
  </xs:choice>
</xs:complexType>
```

The following table lists the child elements of the **ArrayOfGroupedItemsType** complex type.

Element name	Type	Description
GroupedItems	**t:GroupedItemsType** (section 2.2.4.13)	Specifies an array of items that are returned by the **FindItem** operation.

2.2.4.5 t:BasePagingType Complex Type

The **BasePagingType** complex type specifies the base type for derived types that specify paged views.

```
<xs:complexType name="BasePagingType"
  abstract="true"
>
  <xs:attribute name="MaxEntriesReturned"
    type="xs:int"
    use="optional"
   />
</xs:complexType>
```

The following table lists the attribute definitions for the **BasePagineType** complex type.

Attribute name	Type	Description
MaxEntriesReturned	**xs:int** [XMLSCHEMA2]	Specifies the maximum number of entries that are returned with each page of the response. This attribute can be specified.

2.2.4.6 t:ContainsExpressionType Complex Type

The **ContainsExpressionType** complex type specifies a search expression that determines whether a given property contains the supplied constant string value.

```
<xs:complexType name="ContainsExpressionType">
  <xs:complexContent>
    <xs:extension
      base="t:SearchExpressionType"
    >
      <xs:sequence>
        <xs:element
          ref="t:Path"
        />
        <xs:element name="Constant"
          type="t:ConstantValueType"
        />
      </xs:sequence>
      <xs:attribute name="ContainmentMode"
        type="t:ContainmentModeType"
        use="optional"
      />
      <xs:attribute name="ContainmentComparison"
        type="t:ContainmentComparisonType"
        use="optional"
      />
    </xs:extension>
```

```
    </xs:complexContent>
  </xs:complexType>
```

The following table lists the child elements of the **ContainsExpressionType** complex type.

Element name	Type	Description
Path	**t:Path** ([MS-OXWSCDATA] section 2.2.5.8)	Specifies the property to use in a contains search expression. The **Path** element specifies a substitution group, as specified in [XMLSCHEMA0]. An element that is represented by the **Path** element **substitutionGroup** attribute MUST be present.
Constant	**t:ConstantValueType** ([MS-OXWSCDATA] section 2.2.4.20)	Specifies a constant value for a search restriction.

The following table lists the attribute definitions for the **ContainsExpressionType** complex type.

Attribute name	Type	Description
ContainmentMode	**t:ContainmentModeType** (section 2.2.5.2)	Specifies the boundaries of a search.
ContainmentComparison	**t:ContainmentComparisonType** (section 2.2.5.1)	Specifies whether a search ignores cases and spaces.

2.2.4.7 t:ExcludesType Complex Type

The **ExcludesType** complex type specifies a bitwise mask of a property for an exclude search restriction.

```
<xs:complexType name="ExcludesType">
  <xs:complexContent>
    <xs:extension
      base="t:SearchExpressionType"
    >
      <xs:sequence>
        <xs:element
          ref="t:Path"
        />
        <xs:element name="Bitmask"
          type="t:ExcludesValueType"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **ExcludesType** complex type.

Element name	Type	Description
Path	**t:Path (**[MS-OXWSCDATA] section 2.2.5.8)	Specifies the property to use in an excludes search expression. The **Path** element specifies a substitution group, as specified in [XMLSCHEMA0]. An element that is represented by the **Path** element **substitutionGroup** attribute MUST be present.
Bitmask	**t:ExcludesValueType** (section 2.2.4.8)	Specifies a hexadecimal or decimal mask for an excludes restriction.

An excludes restriction can only be applied to a property that has an integer value.

2.2.4.8 t:ExcludesValueType Complex Type

The **ExcludesValueType** complex type specifies a hexadecimal or decimal mask for a restriction that excludes some results.

```
<xs:complexType name="ExcludesValueType">
  <xs:attribute name="Value"
    type="t:ExcludesAttributeType"
    use="required"
   />
</xs:complexType>
```

The following table lists the attribute definitions for the **ExcludesValueType** complex type.

Attribute name	Type	Description
Value	**t:ExcludesAttributeType** ([MS-OXWSCDATA] section 2.2.3.15)	Specifies a decimal or hexadecimal bitmask for an exclude restriction. This attribute SHOULD be present for an excludes restriction.

2.2.4.9 t:ExistsType Complex Type

The **ExistsType** complex type specifies a search restriction that resolves to true if the supplied property exists on an item. The **ExistsType** complex type extends the **SearchExpressionType** complex type, as specified in section 2.2.4.25.

```
<xs:complexType name="ExistsType">
  <xs:complexContent>
    <xs:extension
      base="t:SearchExpressionType"
    >
      <xs:sequence>
        <xs:element
          ref="t:Path"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **ExistsType** complex type.

Element name	Type	Description
Path	**Path** ([MS-OXWSCDATA] section 2.2.5.8)	Specifies the property to use in an excludes search expression. The **Path** element specifies a substitution group, as specified in [XMLSCHEMA0]. An element that is represented by the **Path** element **substitutionGroup** attribute MUST be present.

2.2.4.10 t:FindFolderParentType Complex Type

The **FindFolderParentType** complex type specifies an array of folders that are returned in the results of a call to the **FindFolder** operation, as specified in section 3.1.4.1.

```
<xs:complexType name="t:FindFolderParentType">
  <xs:sequence>
    <xs:element name="Folders"
      type="t:ArrayOfFoldersType"
     />
  </xs:sequence>
  <xs:attributeGroup
    ref="t:FindResponsePagingAttributes"
   />
</xs:complexType>
```

The following table lists the child elements of the **FindFolderParentType** complex type.

Element name	Type	Description
Folders	**t:ArrayOfFoldersType** ([MS-OXWSFOLD] section 2.2.4.1)	Specifies the folders in the result set.

The following table lists the attribute groups that are defined for the **FindFolderParentType** complex type.

Attribute group name	Description
t:FindResponsePagingAttributes ([MS-OXWSCDATA] section 2.2.8.1)	Specifies zero or more attributes that describe the result set.

2.2.4.11 t:FindItemParentType Complex Type

The **FindItemParentType** complex type specifies the results of a search of a single root folder.

```
<xs:complexType name="FindItemParentType">
  <xs:choice>
    <xs:element name="Items"
      type="t:ArrayOfRealItemsType"
```

```
      />
      <xs:element name="Groups"
        type="t:ArrayOfGroupedItemsType"
      />
    </xs:choice>
    <xs:attributeGroup
      ref="t:FindResponsePagingAttributes"
    />
  </xs:complexType>
```

The following table lists the child elements of the **FindItemParentType** complex type.

Element name	Type	Description
Items	**t:ArrayOfRealItemsType** ([MS-OXWSCDATA] section 2.2.4.8)	Specifies the results of a search in which the items returned are not grouped.
Groups	**t:ArrayOfGroupedItemsType** (section 2.2.4.4)	Specifies the grouped results of a search.

The following table lists the attribute groups that are defined for the **FindItemParentType** complex type.

Attribute group name	Description
t:FindResponsePagingAttributes ([MS-OXWSCDATA] section 2.2.8.1)	Specifies zero or more attributes that describe the result set.

If a **GroupBy** or **DistinguishedGroupBy** element was specified in the **FindItemType** complex type, as specified in section 3.1.4.2.3.2, instance that was sent to the **FindItem** operation, as specified in section 3.1.4.2, any results of the operation are returned in the **Groups** element; otherwise, the results are returned in the **Items** element.

2.2.4.12 t:FractionalPageViewType Complex Type

The **FractionalPageViewType** complex type specifies where a paged view starts and the maximum number of items that are returned. The **FractionalPageViewType** complex type extends the **BasePagingType** complex type, as specified in section 2.2.4.5.

```
<xs:complexType name="FractionalPageViewType">
  <xs:complexContent>
    <xs:extension
      base="t:BasePagingType"
    >
      <xs:attribute name="Denominator"
        type="xs:int"
        use="required"
      />
      <xs:attribute name="Numerator"
        type="xs:int"
        use="required"
      />
    </xs:extension>
```

```
  </xs:complexContent>
</xs:complexType>
```

The following table lists the attributes that are defined for the **FractionalPageViewType** complex type.

Attribute name	Type	Description
Denominator	**xs:int** [XMLSCHEMA2]	Specifies the denominator of the fractional offset from the start of the total number of items in the result set.
Numerator	**xs:int**	Specifies the numerator of the fractional offset from the start of the total number of items in the result set.

2.2.4.13 t:GroupedItemsType Complex Type

The **GroupedItemsType** complex type specifies a collection of items that are the result of a grouped **FindItem** operation, as specified in section 3.1.4.2.

```
<xs:complexType name="GroupedItemsType">
  <xs:sequence>
    <xs:element name="GroupIndex"
      type="xs:string"
     />
    <xs:element name="Items"
      type="t:ArrayOfRealItemsType"
     />
  </xs:sequence>
</xs:complexType>
```

The following table lists the child elements of the **GroupedItemsType** complex type.

Element name	Type	Description
GroupIndex	**xs:string** [XMLSCHEMA2]	Specifies the property value that is used to group the items.
Items	**t:ArrayOfRealItemsType** ([MS-OXWSCDATA] section 2.2.4.8)	Specifies the group of items that correspond to the specified group value.

2.2.4.14 t:IndexedPageViewType Complex Type

The **IndexedPageViewType** complex type specifies how paged item information is returned by the **FindItem** operation, as specified in section 3.1.4.2, or the **FindFolder** operation, as specified in section 3.1.4.1. The **IndexedPageViewType** complex type extends the **BasePagingType** complex type, as specified in section 2.2.4.5.

```
<xs:complexType name="IndexedPageViewType">
  <xs:complexContent>
    <xs:extension
```

```
      base="t:BasePagingType"
    >
      <xs:attribute name="Offset"
        type="xs:ing"
        use="required"
       />
      <xs:attribute name="BasePoint"
        type="t:IndexBasePointType"
        use="required"
       />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the attributes that are defined for the **IndexedPageViewType** complex type.

Attribute name	Type	Description
Offset	**xs:int** [XMLSCHEMA2]	Specifies the offset from the **BasePoint** element. This attribute MUST be specified.
BasePoint	**t:IndexBasePointType** (section 2.2.5.4)	Specifies whether the page of items starts at the beginning or end of the set of items that were found by the search. This attribute MUST be specified.

2.2.4.15 t:IsEqualToType Complex Type

The **IsEqualToType** complex type specifies a search expression that compares a property with either a constant value or another property and evaluates to true if they are equal. The **IsEqualToType** complex type extends the **TwoOperandExpressionType** complex type, as specified in section 2.2.4.29.

```
<xs:complexType name="IsEqualToType">
  <xs:complexContent>
    <xs:extension
      base="t:TwoOperandExpressionType"
     />
  </xs:complexContent>
</xs:complexType>
```

2.2.4.16 t:IsGreaterThanOrEqualToType Complex Type

The **IsGreaterThanOrEqualToType** complex type specifies a search expression that compares a property with either a constant value or another property and evaluates to "true" if the first property is greater than or equal to the second. The **IsGreaterThanOrEqualToType** complex type extends the **TwoOperandExpressionType** complex type, as specified in section 2.2.4.29.

```
<xs:complexType name="IsGreaterThanOrEqualToType">
  <xs:complexContent>
    <xs:extension
```

```
      base="t:TwoOperandExpressionType"
     />
  </xs:complexContent>
</xs:complexType>
```

2.2.4.17 t:IsGreaterThanType Complex Type

The **IsGreaterThanType** complex type specifies a search expression that compares a property with either a constant value or another property and returns "true" if the first property is greater. The **IsGreaterThanType** complex type extends the **TwoOperandExpressionType** complex type, as specified in section 2.2.4.29.

```
<xs:complexType name="IsGreaterThanType">
  <xs:complexContent>
    <xs:extension
      base="t:TwoOperandExpressionType"
     />
  </xs:complexContent>
</xs:complexType>
```

2.2.4.18 t:IsLessThanOrEqualToType Complex Type

The **IsLessThanOrEqualToType** complex type specifies a search expression that compares a property with either a constant value or another property and evaluates to "true" if the first property is less than or equal to the second. The **IsLessThanOrEqualToType** complex type extends the **TwoOperandExpressionType** complex type, as specified in section 2.2.4.29.

```
<xs:complexType name="IsLessThanOrEqualToType">
  <xs:complexContent>
    <xs:extension
      base="t:TwoOperandExpressionType"
     />
  </xs:complexContent>
</xs:complexType>
```

2.2.4.19 t:IsLessThanType Complex Type

The **IsLessThanType** complex type specifies a search expression that compares a property with either a constant value or another property and evaluates to "true" if the first property is less than the second. The **IsLessThanType** complex type extends the **TwoOperandExpressionType** complex type, as specified in section 2.2.4.29.

```
<xs:complexType name="IsLessThanType">
  <xs:complexContent>
    <xs:extension
      base="t:TwoOperandExpressionType"
     />
  </xs:complexContent>
</xs:complexType>
```

2.2.4.20 t:IsNotEqualToType Complex Type

The **IsNotEqualToType** complex type specifies a search expression that compares a property with either a constant value or another property and evaluates to "true" if the values are not the same. The **IsNotEqualToType** complex type extends the **TwoOperandExpressionType** complex type, as specified in section 2.2.4.29.

```
<xs:complexType name="IsNotEqualToType">
  <xs:complexContent>
    <xs:extension
      base="t:TwoOperandExpressionType"
     />
  </xs:complexContent>
</xs:complexType>
```

2.2.4.21 t:MultipleOperandBooleanExpressionType Complex Type

The **MultipleOperandBooleanExpressionType** complex type specifies the base type for search expressions that are formed by two or more Boolean operands. The **MultipleOperandBooleanExpressionType** complex type extends the **SearchExpressionType** complex type, as specified in section 2.2.4.25.

```
<xs:complexType name="SearchExpressionType"
  abstract="true"
>
  <xs:complexContent>
    <xs:extension
      base="t:SearchExpressionType"
    >
      <xs:sequence>
        <xs:element
          minOccurs="1"
          maxOccurs="unbounded"
          ref="t:SearchExpression"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **MultipleOperandBooleanExpressionType** complex type.

Element name	Type	Description
SearchExpression	**t:SearchExpression** (section 2.2.3.13)	Specifies an array of search expressions that represents a set of operands.

2.2.4.22 t:NotType Complex Type

The **NotType** complex type specifies a search expression that negates the Boolean value of the search expression that it contains. The **NotType** complex type extends the **SearchExpressionType** complex type, as specified in section 2.2.4.25.

```
<xs:complexType name="NotType">
  <xs:complexContent>
    <xs:extension
      base="t:SearchExpressionType"
    >
      <xs:sequence>
        <xs:element
          ref="t:SearchExpression"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **NotType** complex type.

Element name	Type	Description
SearchExpression	**t:SearchExpression** (section 2.2.3.13)	Specifies a search expression.

2.2.4.23 t:OrType Complex Type

The **OrType** complex type specifies a search expression that performs a logical **OR** on the search expression that it contains. The result of the **OR** operation is **TRUE** if all of the search expressions that are contained with the **OrType** complex type are true. The **OrType** complex type extends the **MultipleOperandBooleanExpressionType** complex type, as specified in section 2.2.4.21.

```
<xs:complexType name="OrType">
  <xs:complexContent>
    <xs:extension
      base="t:MultipleOperandBooleanExpressionType"
    />
  </xs:complexContent>
</xs:complexType>
```

2.2.4.24 t:RestrictionType Complex Type

The **RestrictionType** complex type specifies a search restriction or query for a **FindItem** operation, as specified in section 3.1.4.2, or a **FindFolder** operation, as specified in section 3.1.4.1.

```
<xs:complexType name="RestrictionType">
  <xs:sequence>
    <xs:element
      ref="t:SearchExpression"
    />
  </xs:sequence>
```

```
</xs:complexType>
```

The following table lists the child elements of the **RestrictionType** complex type.

Element name	Type	Description
SearchExpression	**t:SearchExpression** (section 2.2.3.13)	Specifies the search term that defines the restriction.

2.2.4.25 t:SearchExpressionType Complex Type

The **SearchExpressionType** complex type specifies the base type for all search expressions.

```
<xs:complexType name="SearchExpressionType"
  abstract="true"
 />
```

2.2.4.26 t:SearchFolderType Complex Type

The **SearchFolderType** complex type specifies a representation of a search folder that is contained in a mailbox.

```
<xs:complexType name="SearchFolderType">
  <xs:complexContent>
    <xs:extension
      base="t:FolderType"
    >
      <xs:sequence>
        <xs:element name="SearchParameters"
          type="t:SearchParametersType"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **SearchFolderType** complex type.

Element name	Type	Description
SearchParameters	**t:SearchParametersType** (section 2.2.4.27)	Specifies the search parameters that define the contents of the search folder. This element can be present.

2.2.4.27 t:SearchParametersType Complex Type

The **SearchParametersType** complex type specifies the search parameters that define the contents of a search folder.

```
<xs:complexType name="SearchParametersType">
  <xs:sequence>
    <xs:element name="Restriction"
      type="t:RestrictionType"
     />
    <xs:element name="BaseFolderIds"
      type="t:NonEmptyArrayOfBaseFolderIdsType"
     />
  </xs:sequence>
  <xs:attribute name="Traversal"
    type="t:SearchFolderTraversalType"
   />
</xs:complexType>
```

The following table lists the child elements of the **SearchParametersType** complex type.

Element name	Type	Description
Restriction	**t:RestrictionType** (section 2.2.4.24)	Specifies the query that defines the contents of a search folder.
BaseFolderIds	**t:NonEmptyArrayOfBaseFolderIdsType** (section 3.1.4.2.3.8)	Specifies an array of folder identifiers that identify the folders that are searched.

The following table lists the attributes that are defined for the **SearchParametersType** complex type.

Attribute name	Type	Description
Traversal	**t:SearchFolderTraversalType** (section 2.2.5.6)	Specifies the depth of a search folder subtree traversal.

2.2.4.28 t:SeekToConditionPageViewType Complex Type

Note *All of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure.*

The **SeekToConditionPageViewType** complex type specifies a condition that has to be satisfied in order to include an item in the result set from the **FindItem** operation, as specified in section 3.1.4.2, or the **FindFolder** operation, as specified in section 3.1.4.1. The **SeekToConditionPageViewType** complex type extends the **BasePagingType** complex type, as specified in section 2.2.4.5.<2>

```
<xs:complexType name="SeekToConditionPageViewType">
  <xs:complexContent>
    <xs:extension base="t:BasePagingType">
      <xs:sequence>
        <xs:element name="Condition" type="t:RestrictionType"
          minOccurs="1"
        />
      </xs:sequence>
      <xs:attribute name="BasePoint" type="t:IndexBasePointType"
```

```
          use="required"
        />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

The following table lists the child elements of the **SeekToConditionPageViewType** complex type.

Element name	Type	Description
Condition	**t:RestrictionType** (section 2.2.4.24)	Specifies the search term used to determine if an item is to be included in the result set.

The following table lists the attributes of the **SeekToConditionPageViewType** complex type.

Attribute name	Type	Description
BasePoint	**t:IndexBasePointType** (section 2.2.5.4)	Specifies whether the search term should be applied from the beginning or end of the result set.

2.2.4.29 t:TwoOperandExpressionType Complex Type

The **TwoOperandExpressionType** complex type specifies the base type for derived classes that represent a restriction that is formed by comparing two values against one another. The **TwoOperandExpressionType** complex type extends the **SearchExpressionType** complex type, as specified in section 2.2.4.25.

```
<xs:complexType name="TwoOperandExpressionType"
  abstract="true"
>
  <xs:complexContent>
    <xs:extension
      base="t:SearchExpressionType"
    >
      <xs:sequence>
        <xs:element
          ref="t:Path"
        />
        <xs:element name="FieldURIOrConstant"
          type="t:FieldURIOrConstantType"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **TwoOperandExpressionType** complex type.

Element name	Type	Description
Path	**t:Path** ([MS-OXWSCDATA] section	Specifies the property path that is

Element name	Type	Description
	2.2.5.8)	searched on for each item or folder in a search.
FieldURIOrConstant	**t:FieldURIOrConstantType** ([MS-OXWSCDATA] section 2.2.4.30)	Specifies the property or constant that is compared with each item or folder in a restriction.

2.2.5 Simple Types

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

Simple type name	Description
ContainmentComparisonType	Specifies whether a search is exact or whether it ignores casing and spaces.
ContainmentModeType	Specifies the search boundaries.
FolderQueryTraversalType	Specifies the types of subtree traversals for deletion and enumeration.
IndexBasePointType	Specifies whether a page of items that are returned starts at the beginning or at the end of the set of items that are found by the search.
ItemQueryTraversalType	Specifies whether a search finds items in folders or in the **Deleted Items folder**.
SearchFolderTraversalType	Specifies the options for how a folder hierarchy is searched when the contents of a search folder are identified.

2.2.5.1 t:ContainmentComparisonType Simple Type

The **ContainmentComparisonType** simple type specifies whether a search is exact or whether it ignores casing and spaces.

```
<xs:simpleType name="ContainmentComparisonType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="Exact"
     />
    <xs:enumeration
      value="IgnoreCase"
     />
    <xs:enumeration
      value="IgnoreCaseAndNonSpacingCharacters"
     />
    <xs:enumeration
      value="IgnoreNonSpacingCharacters"
     />
    <xs:enumeration
      value="Loose"
```

```
      />
      <xs:enumeration
        value="LooseAndIgnoreCase"
      />
      <xs:enumeration
        value="LooseAndIgnoreCaseAndIgnoreNonSpace"
      />
      <xs:enumeration
        value="LooseAndIgnoreNonSpace"
      />
    </xs:restriction>
  </xs:simpleType>
```

The following table lists the values that are defined by the **ContainmentComparisonType** simple type.

Value	Meaning
Exact	Specifies that the comparison must be exact.
IgnoreCase	Specifies that the comparison ignores casing.
IgnoreCaseAndNonSpacingCharacters	Specifies that the comparison ignores casing and non-spacing characters.
IgnoreNonSpacingCharacters	Specifies that the comparison ignores non-spacing characters.
Loose	This value MUST NOT be used.
LooseAndIgnoreCase	This value MUST NOT be used.
LooseAndIgnoreCaseAndIgnoreNonSpace	This value MUST NOT be used.
LooseAndIgnoreNonSpace	This value MUST NOT be used.

2.2.5.2 t:ContainmentModeType Simple Type

The **ContainmentModeType** simple type specifies the search boundaries.

```
  <xs:simpleType name="ContainmentModeType">
    <xs:restriction>
      <xs:enumeration
        value="ExactPhrase"
      />
      <xs:enumeration
        value="FullString"
      />
      <xs:enumeration
        value="Prefixed"
      />
      <xs:enumeration
        value="PrefixOnWords"
      />
      <xs:enumeration
        value="Substring"
      />
```

```
  </xs:restriction>
</xs:simpleType>
```

The following table lists the values that are defined by the **ContainmentModeType** simple type.

Value	Meaning
ExactPhrase	Specifies that the comparison is between the exact phrase in the property and the constant. If the phrase and the supplied constant are the same, the expression resolves to true.
FullString	Specifies that the comparison is between the full string value of the property and the constant. If the property value and the supplied constant are the same, the expression resolves to true.
Prefixed	Specifies that the comparison is between the prefix of the property and the constant. If the prefix of the property value matches the value that is provided in the constant, the expression resolves to true.
PrefixOnWords	Specifies that the comparison is between a prefix on any individual word in the property value and the constant. If any of the words are prefixed with a value that matches the value that is provided in the constant, the expression resolves to true.
Substring	Specifies that the comparison is between a substring of the property value and the constant. If the substring exists anywhere in the property value, the expression resolves to true.

2.2.5.3 t:FolderQueryTraversalType Simple Type

The **FolderQueryTraversalType** simple type specifies the types of subtree traversals for deletion and enumeration.

```
<xs:simpleType name="FolderQueryTraversalType">
  <xs:restriction>
    <xs:enumeration
      value="Deep"
     />
    <xs:enumeration
      value="Shallow"
     />
    <xs:enumeration
      value="SoftDeleted"
     />
  </xs:restriction>
</xs:simpleType>
```

The following table lists the values that are defined by the **FolderQueryTraversalType** simple type.

Value	Meaning
Deep	Specifies a search in all subfolders of the identified parent folder and returns only the folder IDs for items that have not been deleted.

Value	Meaning
Shallow	Specifies a search in only the identified folder and returns only the folder IDs for items that have not been deleted.
SoftDeleted	Specifies a shallow traversal search for items that are in the Deleted Items folder.

2.2.5.4 t:IndexBasePointType Simple Type

The **IndexBasePointType** simple type specifies whether a page of items that are returned by the **FindFolder** operation, as specified in section 3.1.4.1, or **FindItem** operation, as specified in section 3.1.4.2, start at the beginning or at the end of the set of items that are found by the search.

```
<xs:simpleType name="IndexBasePointType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="Beginning"
     />
    <xs:enumeration
      value="End"
     />
  </xs:restriction>
</xs:simpleType>
```

The following table lists the values that are defined by the **IndexBasePointType** simple type.

Value	Meaning
Beginning	Specifies that the page of items starts from the beginning of the set of items that are returned by the search.
End	Specifies that the page of items starts from the end of the set of items that are returned by the search.

2.2.5.5 t:ItemQueryTraversalType Simple Type

The **ItemQueryTraversalType** simple type specifies whether the search finds items in folders or in the Deleted Items folder.

```
<xs:simpleType name="ItemQueryTraversalType">
  <xs:restriction>
    <xs:enumeration
      value="Shallow"
     />
    <xs:enumeration
      value="SoftDeleted"
     />
  </xs:restriction>
</xs:simpleType>
```

The following table lists the values that are defined by the **ItemQueryTraversalType** simple type.

Value	Description
Shallow	Specifies that only the items in the folder are returned.
SoftDeleted	Specifies that only the items that are in the Deleted Items folder are returned.

2.2.5.6 t:SearchFolderTraversalType Simple Type

The **SearchFolderTraversalType** simple type specifies the options for how a folder hierarchy is searched when the contents of a search folder are identified.

```
<xs:simpleType name="SearchFolderTraversalType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="Deep"
     />
    <xs:enumeration
      value="Shallow"
     />
  </xs:restriction>
</xs:simpleType>
```

The following table lists the values that are defined by the **SearchFolderTraversalType** simple type.

Value	Meaning
Deep	Specifies that a deep search is used to traverse the folder hierarchy.
Shallow	Specifies that a shallow search is used to traverse the folder hierarchy.

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3 Protocol Details

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

3.1 ExchangeServicePortType Server Details

The Mailbox Search Web Service Protocol defines a single port type with two operations. These operations enable client applications to search for items in the mailbox and folders.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

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

Operation name	Description
FindFolder	Searches the data store and returns a folder or folders that match the specified search criteria.
FindItem	Searches the data store and returns the item or items that match the specified search criteria.

3.1.4.1 FindFolder Operation

The **FindFolder** operation obtains a list of folders that meet specified search criteria by searching the subfolders of a specified folder.

The following is the WSDL port type specification of the operation.

```
<wsdl:operation name="FindFolder">
     <wsdl:input message="tns:FindFolderSoapIn"/>
     <wsdl:output message="tns:FindFolderSoapOut"/>
</wsdl:operation>
```

The following is the WSDL binding specification of the operation.

```
<wsdl:operation name="FindFolder">
```

```
   <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/FindFolder"/>
   <wsdl:input>
      <soap:header message="tns:FindFolderSoapIn" part="Impersonation" use="literal"/>
      <soap:header message="tns:FindFolderSoapIn" part="MailboxCulture" use="literal"/>
      <soap:header message="tns:FindFolderSoapIn" part="RequestVersion" use="literal"/>
      <soap:header message="tns:FindFolderSoapIn" part="TimeZoneContext" use="literal"/>
      <soap:body parts="request" use="literal" />
   </wsdl:input>
   <wsdl:output>
      <soap:body parts="FindFolderResult" use="literal" />
      <soap:header message="tns:FindFolderSoapOut" part="ServerVersion" use="literal"/>
   </wsdl:output>
</wsdl:operation>
```

3.1.4.1.1 Messages

The following table lists the WSDL message definitions that are specific to this operation.

Message name	Description
FindFolderSoapIn	Specifies the **SOAP message** that defines the folder or folders to search for.
FindFolderSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.1.1.1 tns:FindFolderSoapIn Message

Note *Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.*

The **FindFolderSoapIn** WSDL message specifies the search criteria for the folder or folders that are to be returned by the **FindFolder** operation.

```
<wsdl:message name="FindFolderSoapIn">
   <wsdl:part name="request" element="tns:FindFolder"/>
   <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
   <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
   <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
   <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext"/>
   <wsdl:part name="ManagementRole" element="t:ManagementRole" />
</wsdl:message>
```

The **FindFolderSoapIn** WSDL message is the input message for the **SOAP action** http://schemas.microsoft.com/exchange/services/2006/messages/FindFolder.

The parts of the **FindFolderSoapIn** WSDL message are described in the following table.

Part name	Element/type	Description
request	**tns:FindFolder** (section 3.1.4.1.2.1)	Specifies the **SOAP body** of the request.
Impersonation	**t:ExchangeImpersonation** ([MS-	Specifies a **SOAP header** that identifies the user whom the client application is

Part name	Element/type	Description
	OXWSCDATA] section 2.2.5.3)	impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.5.6)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.5.9)	Specifies a SOAP header that identifies the schema version for the **FindFolder** operation request.
TimeZoneContext	**t:TimeZoneContext** ([MS-OXWSGTZ] section 2.2.3.4)	Specifies a SOAP header that identifies the time zone to use for all responses from the server. All times that are returned from the server will be converted to the specified time zone.
ManagementRole	**t:ManagementRole** ([MS-OXWSCDATA] section 2.2.5.7)	Specifies a SOAP header that identifies the server roles that are necessary in order for the caller to make the request.<3>

3.1.4.1.1.2 tns:FindFolderSoapOut Message

The **FindFolderSoapOut** WSDL message specifies the server response to the **FindFolder** operation request to find a folder or folders.

```
<wsdl:message name="FindFolderSoapOut">
   <wsdl:part name="FindFolderResult" element="tns:FindFolderResponse"/>
   <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

The **FindFolderSoapOut** WSDL message is the output message for the SOAP action http://schemas.microsoft.com/exchange/services/2006/messages/FindFolder.

The parts of the **FindFolderSoapOut** WSDL message are described in the following table.

Part name	Element/type	Description
FindFolderResult	**tns:FindFolderResponse** (section 3.1.4.1.2.2)	Specifies SOAP body to a **FindFolder** operation request..
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.5.10)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.1.2 Elements

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

Element name	Description
FindFolder	Specifies the criteria required to search for a folder or folders.
FindFolderResponse	Specifies the response body from a request to search for a folder or folders.

3.1.4.1.2.1 tns:FindFolder Element

The **FindFolder** element specifies the base element for a **FindFolder** operation request.

```
<xs:element name="FindFolder"
  type="m:FindFolderType"
 />
```

3.1.4.1.2.2 tns:FindFolderResponse Element

The **FindFolderResponse** element specifies the response message for a **FindFolder** operation.

```
<xs:element name="FindFolderResponse"
  type="m:FindFolderResponseType"
 />
```

3.1.4.1.3 Complex Types

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

Complex type name	Description
FindFolderResponseType	Specifies the response from the **FindFolder** operation.
FindFolderType	Specifies the criteria for searching for a folder or folders with the **FindFolder** operation.

3.1.4.1.3.1 m:FindFolderResponseType Complex Type

The **FindFolderResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.4.16).

```
<xs:complexType name="FindFolderResponseType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseResponseMessageType"
     />
  </xs:complexContent>
</xs:complexType>
```

3.1.4.1.3.2 m:FindFolderType Complex Type

The **FindFolderType** complex type specifies a request to find folders in a mailbox. The **FindFolderType** complex type extends the **BaseRequestType** complex type ([MS-OXWSCDATA] section 2.2.4.15).

```
<xs:complexType name="FindFolderType">
  <xs:complexContent>
    <xs:extension
```

```
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="FolderShape"
          type="t:FolderResponseShapeType"
         />
        <xs:choice
          maxOccurs="1"
          minOccurs="01"
        >
          <xs:element name="IndexedPageFolderView"
            type="t:IndexedPageViewType"
           />
          <xs:element name="FractionalPageFolderView"
            type="t:FractionalPageViewType"
           />
        </xs:choice>
        <xs:element name="Restriction"
          type="t:RestrictionType"
          minOccurs="0"
         />
        <xs:element name="ParentFolderIds"
          type="t:NonEmptyArrayOfBaseFolderIdsType"
         />
      </xs:sequence>
      <xs:attribute name="Traversal"
        type="t:FolderQueryTraversalType"
        use="required"
       />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **FindFolderType** complex type.

Element name	Type	Description
FolderShape	**t:FolderResponseShapeType** ([MS-OXWSCDATA] section 2.2.4.32)	Specifies the contents of the query response.
IndexedPageFolderView	**t:IndexedPageViewType** (section 2.2.4.14)	Specifies how paged information is returned by the query.
FractionalPageFolderView	**t:FractionalPageViewType** (section 2.2.4.12)	Specifies the starting item and the number of items that are returned by a paged query.
Restriction	**t:RestrictionType** (section 2.2.4.24)	Specifies the search parameters that define the folder query.
ParentFolderIds	**t:NonEmptyArrayOfBaseFolderIdsType** ([MS-OXWSFOLD] section 3.1.4.6.3.3)	Specifies the folders that the query searches.

The following table lists the attributes that are defined for the **FindFolderType** complex type.

Attribute name	Type	Description
Traversal	**t:FolderQueryTraversalType** (section 2.2.5.3)	Specifies the traversal scheme that is used to search for folders.

The **FindFolderType** complex type specifies the folders to search for as well as the structure of the response.

Either the **IndexedPageFolderView** or the **FractionalPageFolderView** element can be specified to paginate the items that are returned in the response. If an element is included, all required subelements MUST be included in the request; if an element is not included, its required subelements MUST NOT be included in the request.

3.1.4.2 FindItem Operation

The **FindItem** operation searches the mailbox and returns items that meet a specified search criteria.

The following is the WSDL port type specification for the operation.

```
<wsdl:operation name="FindItem">
     <wsdl:input message="tns:FindItemSoapIn"/>
     <wsdl:output message="tns:FindItemSoapOut"/>
</wsdl:operation>
```

The following is the WSDL binding specification for the operation.<4>

```
<wsdl:operation name="FindItem">
   <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/FindItem"/>
   <wsdl:input>
      <soap:header message="tns:FindItemSoapIn" part="Impersonation" use="literal"/>
      <soap:header message="tns:FindItemSoapIn" part="MailboxCulture" use="literal"/>
      <soap:header message="tns:FindItemSoapIn" part="RequestVersion" use="literal"/>
      <soap:header message="tns:FindItemSoapIn" part="TimeZoneContext" use="literal"/>
      <soap:header message="tns:FindItemSoapIn" part="DateTimePrecision" use="literal" />
      <soap:body parts="request" use="literal" />
   </wsdl:input>
   <wsdl:output>
      <soap:body parts="FindItemResult" use="literal" />
      <soap:header message="tns:FindItemSoapOut" part="ServerVersion" use="literal"/>
   </wsdl:output>
</wsdl:operation>
```

3.1.4.2.1 Messages

The following table lists the WSDL message definitions that are specific to this operation.

Message name	Description
FindItemSoapIn	Specifies the SOAP message that contains the criteria required to search for items.

Message name	Description
FindItemSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.2.1.1 tns:FindItemSoapIn Message

Note *Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.*

The **FindItemSoapIn** WSDL message specifies the search criteria for the items or items that the **FindItem** operation returns.

```
<wsdl:message name="FindItemSoapIn">
   <wsdl:part name="request" element="tns:FindItem"/>
   <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
   <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
   <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
   <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext"/>
   <wsdl:part name="DateTimePrecision" element="t:DateTimePrecision" />
   <wsdl:part name="ManagementRole" element="t:ManagementRole" />
</wsdl:message>
```

The **FindItemSoapIn** WSDL message is the input message for the SOAP action http://schemas.microsoft.com/exchange/services/2006/messages/FindItem.

The parts of the **FindItemSoapIn** WSDL message are described in the following table.

Part name	Element/type	Description
request	**tns:FindItem** (section 3.1.4.2.2.1)	Specifies SOAP body of the request.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.5.3)	Specifies a SOAP header that identifies the user that the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.5.6)	Specifies a SOAP header that identifies the culture to use to access the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.5.9)	Specifies a SOAP header that identifies the schema version for the **FindItem** operation request.
TimeZoneContext	**t:TimeZoneContext** ([MS-OXWSGTZ] section 2.2.3.4)	Specifies a SOAP header that identifies the time zone to use for all responses from the server. All times that are returned from the server will be converted to the specified time zone.
DateTimePrecision	**t:DateTimePrecision** ([MS-OXWSCDATA] section 2.2.5.1)	Specifies a SOAP header that identifies the resolution of date/time values in responses from the server, either in seconds or in milliseconds.<5>
ManagementRole	**t:ManagementRole** ([MS-	Specifies a SOAP header that identifies the

Part name	Element/type	Description
	OXWSCDATA] section 2.2.5.7)	server roles that are necessary in order for the caller to make the request.<6>

3.1.4.2.1.2 tns:FindItemSoapOut Message

The **FindItemSoapOut** WSDL message specifies the server response to the **FindItem** operation request.

```
<wsdl:message name="FindItemSoapOut">
   <wsdl:part name="FindItemResult" element="tns:FindItemResponse"/>
   <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

The **FindItemSoapOut** WSDL message is the output message for the SOAP action http://schemas.microsoft.com/exchange/services/2006/messages/FindItem.

The parts of the **FindItemSoapOut** WSDL message are described in the following table.

Part name	Element/type	Description
FindItemResult	**tns:FindItemResponse** (section 3.1.4.2.2.2)	Specifies SOAP body of the response to a **FindItem** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.5.10)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.2.2 Elements

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

Element name	Description
FindItem	Specifies the base element for the FindItem operation.
FindItemResponse	Specifies the response element for the FindItem operation.

3.1.4.2.2.1 tns:FindItem Element

The **FindItem** element specifies the base element for a **FindItem** operation.

```
<xs:element name="FindItem"
  type="m:FindItemType"
 />
```

3.1.4.2.2.2 tns:FindItemResponse Element

The **FindItemResponse** element specifies the response message for the **FindItem** operation.

```
<xs:element name="FindItemResponse"
  type="m:FindItemResponseType"
```

```
/>
```

3.1.4.2.3 Complex Types

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

Complex type name	Description
FindItemResponseType	Specifies the response to the FindItem operation request.
FindItemType	Specifies the request to the FindItem operation.
AggregateOnType	Specifies the property that is used to determine the order of grouped items for a grouped result set.
BaseGroupBy	Specifies the base class for derived complex types that specify grouped queries to the FindItem and FindFolder operations.
DistinguishedGroupByType	Specifies a standard grouping for the FindItem operation.
FieldOrderType	Specifies a single field by which to sort results and specifies the direction of the sort.
GroupByType	Specifies the grouping for items returned by the FindItem operation.
NonEmptyArrayOfFieldOrdersType	Specifies an array of FieldOrderType complex type elements that contains at least one member.

3.1.4.2.3.1 m:FindItemResponseType Complex Type

The **FindItemResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.4.16).

```
<xs:complexType name="FindItemResponseType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseResponseMessageType"
     />
  </xs:complexContent>
</xs:complexType>
```

3.1.4.2.3.2 m:FindItemType Complex Type

The **FindItemType** complex type specifies the search criteria to use for the **FindItem** operation. The **FindItemType** complex type extends the **BaseRequestType** complex type ([MS-OXWSCDATA] section 2.2.4.15).

```
<xs:complexType name="FindItemType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
```

```
        <xs:element name="ItemShape"
          type="t:ItemResponseShapeType"
         />
        <xs:choice
          minOccurs="0"
        >
          <xs:element name="IndexedPageItemView"
            type="t:IndexedPageViewType"
           />
          <xs:element name="FractionalPageItemView"
            type="t:FractionalPageViewType"
           />
          <xs:element name="CalendarView"
            type="t:CalendarViewType"
           />
          <xs:element name="ContactsView"
            type="t:ContactsViewType"
           />
        </xs:choice>
        <xs:choice
          minOccurs="0"
        >
          <xs:element name="GroupBy"
            type="t:GroupByType"
           />
          <xs:element name="DistinguishedGroupBy"
            type="t:DistinguishedGroupByType"
           />
        </xs:choice>
        <xs:element name="Restriction"
          type="t:RestrictionType"
          minOccurs="0"
         />
        <xs:element name="SortOrder"
          type="t:NonEmptyArrayOfFieldOrdersType"
          minOccurs="0"
         />
        <xs:element name="ParentFolderIds"
          type="t:NonEmptyArrayOfBaseFolderIdsType"
          minOccurs="0"
         />
        <xs:element name="QueryString"
          type="xs:string"
          minOccurs="0"
         />
      </xs:sequence>
      <xs:attribute name="Traversal"
        type="t:ItemQueryTraversalType"
       />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **FindItemType** complex type

Element name	Type	Description
ItemShape	**t:ItemResponseShapeType** ([MS-OXWSCDATA] section 2.2.4.38)	Specifies the array of items that are returned by the query.
IndexedPageItemView	**t:IndexedPageViewType** (section 2.2.4.14)	Specifies how paged item information is returned in the response.
FractionalPageItemView	**t:FractionalPageViewType** (section 2.2.4.12)	Specifies the starting item and number of items to return by the query.
CalendarView	**t:CalendarViewType** ([MS-OXWSMTGS] section 2.2.4.7)	Specifies the settings that are used to return calendar items as they appear in a calendar.
ContactsView	**t:ContactsViewType** ([MS-OXWSCONT] section 3.1.4.1.2.3)	Specifies the settings that are used to return contact items based on their alphabetical display names.
GroupBy	**t:GroupByType** (section 3.1.4.2.3.7)	Specifies the grouping for items that are returned by a query.
DistinguishedGroupBy	**t:DistinguishedGroupByType** (section 3.1.4.2.3.5)	Specifies a standard grouping.
Restriction	**t:RestrictionType** (section 2.2.4.24)	Specifies a search restriction or query.
SortOrder	**t:NonEmptyArrayOfFieldOrdersType** (section 3.1.4.2.3.8)	Specifies one or more **FieldOrderType** complex type (section 3.1.4.2.3.6) elements that specify how the results should be sorted.
ParentFolderIds	**t:NonEmptyArrayOfBaseFolderIdsType** ([MS-OXWSFOLD] section 3.1.4.6.3.3)	Specifies one or more folders that are the root of the search.
QueryString	**xs:string** ([XMLSCHEMA2])	Specifies the query that is used for the search.

The following table lists the attributes that are defined for the **FindItemType** complex type.

Attribute name	Type	Description
Traversal	**t:ItemQueryTraversalType** (section 2.2.5.5)	Specifies whether the search finds items in folders or in the Deleted Items folder.

The **FindItemType** complex type specifies the search criteria to find a set of items by using the **FindItem** operation as well as the structure of the response.

One of the following elements can be included in the request to specify how the returned items are viewed. If an element is included, all required subelements MUST be included in the request; if an element is not included, its required subelements MUST NOT be included in the request.

- **IndexedPageItemView**
- **FractionalPageItemView**
- **CalendarView**
- **ContactsView**

One of the following elements can be included in the request to specify how the results, if any, are to be grouped:

- **GroupBy**
- **DistinguishedGroupBy**

When the **ItemShape** element is set to "All" or "Default", the properties that are returned by the **FindItem** operation depend on the folder that is searched for the items. The properties that are returned are defined by the complex type element or combination of complex type elements that represent the item stored in the folder, as shown in the following table.<7>

Folder name	Complex type
Calendar	**CalendarItemType** ([MS-OXWSMTGS] section 2.2.4.4).
Contacts	**ContactItemType** ([MS-OXWSCONT] section 2.2.4.2). **t:DistributionListType** ([MS-OXWSDLIST] section 2.2.4.3).
Folder	**MessageType** ([MS-OXWSMSG] section 2.2.4.1). **PostItemType** ([MS-OXWSPOST] section 2.2.4.1). **MeetingCancellationMessageType** ([MS-OXWSMTGS] section 2.2.4.11). **MeetingMessageType** ([MS-OXWSMTGS] section 2.2.4.12). **MeetingRequestMessageType** [MS-OXWSMTGS] section 2.2.4.13). **MeetingResponseMessageType** [MS-OXWSMTGS] section 2.2.4.14).
Search	**CalendarItemType** ([MS-OXWSMTGS] section 2.2.4.4). **ContactItemType** ([MS-OXWSCONT] section 2.2.4.2). **DistributionListType** ([MS-OXWSDLIST] section 2.2.4.3). **MessageType** ([MS-OXWSMSG] section 2.2.4.1). **PostItemType** ([MS-OXWSPOST] section 2.2.4.1). **MeetingCancellationMessageType** [MS-OXWSMTGS] section 2.2.4.11). **MeetingMessageType** ([MS-OXWSMTGS] section 2.2.4.12). **MeetingRequestMessageType** ([MS-OXWSMTGS] section 2.2.4.13). **MeetingResponseMessageType** ([MS-OXWSMTGS] section 2.2.4.14). **TaskType** [MS-OXWSTASK] section 2.2.4.3).
Tasks	**TaskType** ([MS-OXWSTASK] section 2.2.4.3).

3.1.4.2.3.3 t:AggregateOnType Complex Type

The AggregateOnType complex type specifies the property that is used to determine the order of grouped items for a grouped result set. When an **AggregateOnType** complex type element is specified, one of the following child elements MUST be specified:

- **FieldURI**
- **IndexedFieldURI**
- **ExtendedFieldURI**

```
<xs:complexType name="AggregateOnType">
  <xs:choice>
    <xs:element name="FieldURI"
      type="t:PathToUnindexedFieldType"
     />
    <xs:element name="IndexedFieldURI"
      type="t:PathToIndexedFieldType"
     />
    <xs:element name="ExtendedFieldURI"
      type="t:PathToExtendedFieldType"
     />
  </xs:choice>
  <xs:attribute name="Aggregate"
    type="t:AggregateType"
    use="required"
   />
</xs:complexType>
```

The following table lists the child elements of the **AggregateOnType** complex type.

Element name	Type	Description
FieldURI	**t:PathToUnindexedFieldType** ([MS-OXWSCDATA] section 2.2.4.49)	Specifies a well-known data store property that is used to group the items.
IndexedFieldURI	**t:PathToIndexedFieldType** ([MS-OXWSCDATA] section 2.2.4.48)	Specifies an individual member of a dictionary that is used to group the items.
ExtendedFieldURI	**t:PathToExtendedFieldType** ([MS-OXWSXPROP] section 2.1.6)	Specifies an extended property that is used to group the items.

The following table lists the attributes that are defined for the **AggregateOnType** complex type.

Attribute name	Type	Description
Aggregate	**t:AggregateType** (section 3.1.4.2.4.1)	Specifies the value that indicates whether the maximum or minimum value of the property specified is used for ordering a group of items. This attribute MUST be set.

3.1.4.2.3.4 t:BaseGroupByType Complex Type

The **BaseGroupByType** complex type specifies the base class for derived complex types that specify grouped queries to the FindItem operation, as specified in section 3.1.4.2, and the FindFolder operation, as specified in section 3.1.4.1.

```
<xs:complexType name="BaseGroupByType"
  abstract="true"
>
  <xs:attribute name="Order"
    type="t:SortDirectionType"
   />
</xs:complexType>
```

The following table lists the attributes that are defined for the **BaseGroupByType** complex type.

Attribute name	Type	Description
Order	**t:SortDirectionType** (section 3.1.4.2.4.2)	Specifies the sort order of the returned grouped items. This attribute MUST be specified.

3.1.4.2.3.5 t:DistinguishedGroupByType Complex Type

The DistinguishedGroupByType complex type specifies a standard grouping for the FindItem operation. The **DistinguishedGroupByType** complex type extends the BaseGroupByType complex type, as specified in section 3.1.4.2.3.4.

```
<xs:complexType name="DistinguishedGroupByType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseGroupByType"
    >
      <xs:sequence>
        <xs:element name="StandardGroupBy"
          type="t:StandardGroupByType"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **DistinguishedGroupByType** complex type.

Element name	Type	Description
StandardGroupBy	**t:StandardGroupByType** (section 3.1.4.2.4.3)	Specifies one of the standard groupings for returned items.

3.1.4.2.3.6 t:FieldOrderType Complex Type

The **FieldOrderType** complex type specifies a single field by which to sort results and specifies the direction of the sort.

```
<xs:complexType name="FieldOrderType">
  <xs:sequence>
    <xs:element
      ref="t:Path"
     />
  </xs:sequence>
  <xs:attribute name="Order"
    type="t:SortDirectionType"
   />
</xs:complexType>
```

The following table lists the child elements of the **FieldOrderType** complex type.

Element name	Type	Description
Path	**t:Path** ([MS-OXWSCDATA] section 2.2.5.8)	Specifies the **URI** that describes the field by which the results are sorted.

The following table lists the attributes that are defined for the **FieldOrderType** complex type.

Attribute name	Type	Description
Order	**t:SortDirectionType** (section 3.1.4.2.4.2)	Specifies the direction of the sort. This attribute MUST be specified.

3.1.4.2.3.7 t:GroupByType Complex Type

The **GroupByType** complex type specifies the grouping for items that are returned by the **FindItem** operation. The **GroupByType** complex type extends the **BaseGroupByType** complex type, as specified in section 3.1.4.2.3.4.

```
<xs:complexType name="GroupByType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseGroupByType"
    >
      <xs:sequence>
       <xs:choice>
         <xs:element name="FieldURI"
           type="t:PathToUnindexedFieldType"
          />
         <xs:element name="IndexedFieldURI"
           type="t:PathToIndexedFieldType"
          />
         <xs:element name="ExtendedFieldURI"
           type="t:PathToExtendedFieldType"
          />
```

```
        </xs:choice>
        <xs:element name="AggregateOn"
          type="t:AggregateOnType"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **GroupByType** complex type.

Element name	Type	Description
FieldURI	**t:PathToUnindexedFieldType** ([MS-OXWSCDATA] section 2.2.4.49)	Specifies the URI to an unindexed item property.
IndexedFieldURI	**t:PathToIndexedFieldType** ([MS-OXWSCDATA] section 2.2.4.48)	Specifies the URI to an indexed item property.
ExtendedFieldURI	**t:PathToExtendedFieldType** ([MS-OXWSXPROP] section 2.1.6)	Specifies an extended item property.
AggregateOn	**t:AggregateOnType** (section 3.1.4.2.3.3)	Specifies the item property that is used to determine the order of groups in a response.

3.1.4.2.3.8 t:NonEmptyArrayOfFieldOrdersType Complex Type

The NonEmptyArrayOfFieldOrdersType complex type specifies an array of **FieldOrderType** complex type, as specified in section 3.1.4.2.3.6, elements that contains at least one member.

```
<xs:complexType name="NonEmptyArrayOfFieldOrdersType">
  <xs:sequence>
    <xs:element name="FieldOrder"
      type="t:FieldOrderType"
      maxOccurs="unbounded"
    />
  </xs:sequence>
</xs:complexType>
```

The following table lists the child elements of the **NonEmptyArraOfFieldOrdersType** complex type.

Element name	Type	Description
FieldOrder	**t:FieldOrderType** (section 3.1.4.2.3.6)	Specifies one or more **FieldOrderType** complex types.

3.1.4.2.4 Simple Types

The following XML schema simple type definitions are specific to this operation.

Simple type name	Description
AggregateType	Specifies the maximum or minimum value used to order items in a group.
SortDirectionType	Specifies the ordering options for a group.
StandardGroupByType	Specifies standard grouping and aggregating mechanisms.

3.1.4.2.4.1 t:AggregateType Simple Type

The **AggregateType** simple type specifies whether the maximum or minimum value of a representative property is used to order the items in a group that is returned by the **FindItem** operation.

```
<xs:simpleType name="AggregateType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="Maximum"
     />
    <xs:enumeration
      value="Minimum"
     />
  </xs:restriction>
</xs:simpleType>
```

The following table lists the values that are defined by the **AggregateType** simple type.

Value	Meaning
Maximum	Specifies that the groups are sorted starting with the maximum value for a specified aggregation property.
Minimum	Specifies that the groups are sorted starting with the minimum value for a specified aggregation property.

3.1.4.2.4.2 t:SortDirectionType Simple Type

The **SortDirectionType** simple type specifies the ordering options for the groups in the grouped item array that is returned in the response.

```
<xs:simpleType name="SortDirectionType">
  <xs:restriction>
    <xs:enumeration
      value="Ascending"
     />
    <xs:enumeration
      value="Descending"
     />
  </xs:restriction>
</xs:simpleType>
```

The following table lists the values that are defined by the **SortDirectionType** simple type.

Value	Meaning
Ascending	Specifies that the items are sorted in ascending order.
Descending	Specifies that the items are sorted in descending order.

3.1.4.2.4.3 t:StandardGroupByType Simple Type

The **StandardGroupByType** simple type specifies the standard grouping and aggregating mechanisms for a grouped response to the FindItem operation.

```
<xs:simpleType name="StandardGroupByType">
  <xs:restriction>
    <xs:enumeration
      value="ConversationTopic"
     />
  </xs:restriction>
</xs:simpleType>
```

The following table lists the values that are defined by the **StandardGroupByType** simple type.

Value	Meaning
ConversationTopic	Specifies that results are grouped by the conversation topic and aggregated on the date and time at which the item was received.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

None.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

The XML files that are listed in the following table are required in order to implement the functionality specified in this document. The contents of each file are included in this section.

File name	Description	Section
MS-OXWSSRCH.wsdl	Contains the WSDL for the implementation of this protocol.	6
MS-OXWSSRCH-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	7.1
MS-OXWSSRCH-types.xsd	Contains the XML schema type definitions that are used in this protocol.	7.2

These files have to be placed in a common folder in order for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSSRCH-types.xsd or MS-OXWSSRCH-messages.xsd schemas have to be placed in the common folder along with the files.

This section contains the contents of the MS-OXWSSRCH.wsdl file.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
  <wsdl:types>
    <xs:schema id="messages" elementFormDefault="qualified" version="Exchange2012"
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"/>
      <xs:include schemaLocation="MS-OXWSSRCH-messages.xsd" />
      <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd" />

    </xs:schema>
    <xs:schema id="types" elementFormDefault="qualified" version="Exchange2012"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>

    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="ExchangeServicePortType">
      <wsdl:operation name="FindFolder">
          <wsdl:input message="tns:FindFolderSoapIn"/>
          <wsdl:output message="tns:FindFolderSoapOut"/>
      </wsdl:operation>
      <wsdl:operation name="FindItem">
          <wsdl:input message="tns:FindItemSoapIn"/>
          <wsdl:output message="tns:FindItemSoapOut"/>
      </wsdl:operation>
```

```
  </wsdl:portType>
  <wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
    <wsdl:documentation>
      <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0" xmlns:wsi="http://ws-
i.org/schemas/conformanceClaim/"/>
    </wsdl:documentation>
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="FindFolder">
            <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/FindFolder"/>
            <wsdl:input>
                <soap:header message="tns:FindFolderSoapIn" part="Impersonation"
use="literal"/>
                <soap:header message="tns:FindFolderSoapIn" part="MailboxCulture"
use="literal"/>
                <soap:header message="tns:FindFolderSoapIn" part="RequestVersion"
use="literal"/>
                <soap:header message="tns:FindFolderSoapIn" part="TimeZoneContext"
use="literal"/>
                <soap:header message="tns:FindFolderSoapIn" part="ManagementRole"
use="literal"/>
                <soap:body parts="request" use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body parts="FindFolderResult" use="literal" />
                <soap:header message="tns:FindFolderSoapOut" part="ServerVersion"
use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="FindItem">
            <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/FindItem"/>
            <wsdl:input>
                <soap:header message="tns:FindItemSoapIn" part="Impersonation"
use="literal"/>
                <soap:header message="tns:FindItemSoapIn" part="MailboxCulture"
use="literal"/>
                <soap:header message="tns:FindItemSoapIn" part="RequestVersion"
use="literal"/>
                <soap:header message="tns:FindItemSoapIn" part="TimeZoneContext"
use="literal"/>
                <soap:header message="tns:FindItemSoapIn" part="DateTimePrecision"
use="literal" />
                <soap:header message="tns:FindItemSoapIn" part="ManagementRole"
use="literal"/>
                <soap:body parts="request" use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body parts="FindItemResult" use="literal" />
                <soap:header message="tns:FindItemSoapOut" part="ServerVersion"
use="literal"/>
            </wsdl:output>
        </wsdl:operation>

  </wsdl:binding>
    <wsdl:message name="FindItemSoapIn">
        <wsdl:part name="request" element="tns:FindItem"/>
        <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
        <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
        <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
```

```
            <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext"/>
             <wsdl:part name="DateTimePrecision" element="t:DateTimePrecision" />
             <wsdl:part name="ManagementRole" element="t:ManagementRole" />
        </wsdl:message>
        <wsdl:message name="FindItemSoapOut">
            <wsdl:part name="FindItemResult" element="tns:FindItemResponse"/>
            <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
        </wsdl:message>

        <wsdl:message name="FindFolderSoapIn">
            <wsdl:part name="request" element="tns:FindFolder"/>
            <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
            <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
            <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
            <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext"/>
            <wsdl:part name="ManagementRole" element="t:ManagementRole" />
        </wsdl:message>
        <wsdl:message name="FindFolderSoapOut">
            <wsdl:part name="FindFolderResult" element="tns:FindFolderResponse"/>
            <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
        </wsdl:message>
    </wsdl:definitions>
```

7 Appendix B: Full XML Schema

For ease of implementation, the following sections provide the full XML schema for this protocol.

Schema name	Prefix	Section
Messages schema	m:	7.1
Types schema	T:	7.2

These files have to be placed in a common folder in order for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSSRCH-types.xsd or MS-OXWSSRCH-messages.xsd schemas have to be placed in the common folder along with the files listed in the table.

7.1 Messages Schema

This section contains the full XML schema for messages.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
elementFormDefault="qualified" version="Exchange2012" id="messages">
    <!--    CHANGE THE SCHEMA LOCATION TO REFLECT THE TYPES XSD ASSOCIATED WITH THIS DOCUMENT
    <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="types.xsd"/>
    <xs:include/>
-->
    <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="MS-OXWSSRCH-types.xsd"/>
    <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
    <xs:complexType name="FindFolderType">
        <xs:complexContent>
            <xs:extension base="m:BaseRequestType">
                <xs:sequence>
                    <xs:element name="FolderShape" type="t:FolderResponseShapeType"/>
                    <xs:choice minOccurs="0">
                        <xs:element name="IndexedPageFolderView"
type="t:IndexedPageViewType"/>
                        <xs:element name="FractionalPageFolderView"
type="t:FractionalPageViewType"/>
                    </xs:choice>
                    <xs:element name="Restriction" type="t:RestrictionType" minOccurs="0"/>
                    <xs:element name="ParentFolderIds"
type="t:NonEmptyArrayOfBaseFolderIdsType"/>
                </xs:sequence>
                <xs:attribute name="Traversal" type="t:FolderQueryTraversalType"
use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="FindFolder" type="m:FindFolderType"/>
    <xs:complexType name="FindFolderResponseMessageType">
        <xs:complexContent>
```

```
            <xs:extension base="m:ResponseMessageType">
                <xs:sequence>
                    <xs:element name="RootFolder" type="t:FindFolderParentType"
minOccurs="0"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="FindFolderResponseType">
        <xs:complexContent>
            <xs:extension base="m:BaseResponseMessageType"/>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="FindFolderResponse" type="m:FindFolderResponseType"/>
    <xs:complexType name="FindItemType">
        <xs:complexContent>
            <xs:extension base="m:BaseRequestType">
                <xs:sequence>
                    <xs:element name="ItemShape" type="t:ItemResponseShapeType"/>
                    <xs:choice minOccurs="0">
                        <xs:element name="IndexedPageItemView" type="t:IndexedPageViewType"/>
                        <xs:element name="FractionalPageItemView"
type="t:FractionalPageViewType"/>
                        <xs:element name="SeekToConditionPageItemView"
type="t:SeekToConditionPageViewType"/>
                        <xs:element name="CalendarView" type="t:CalendarViewType"/>
                        <xs:element name="ContactsView" type="t:ContactsViewType"/>
                    </xs:choice>
                    <xs:choice minOccurs="0">
                        <xs:element name="GroupBy" type="t:GroupByType"/>
                        <xs:element name="DistinguishedGroupBy"
type="t:DistinguishedGroupByType"/>
                    </xs:choice>
                    <xs:element name="Restriction" type="t:RestrictionType" minOccurs="0"/>
                    <xs:element name="SortOrder" type="t:NonEmptyArrayOfFieldOrdersType"
minOccurs="0"/>
                    <xs:element name="ParentFolderIds"
type="t:NonEmptyArrayOfBaseFolderIdsType"/>
                    <xs:element name="QueryString" type="m:QueryStringType" minOccurs="0"
maxOccurs="1"/>
                </xs:sequence>
                <xs:attribute name="Traversal" type="t:ItemQueryTraversalType"
use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="FindItem" type="m:FindItemType"/>
    <xs:complexType name="FindItemResponseMessageType">
        <xs:complexContent>
            <xs:extension base="m:ResponseMessageType">
                <xs:sequence>
                    <xs:element name="RootFolder" type="t:FindItemParentType" minOccurs="0"/>
                    <xs:element name="HighlightTerms" type="t:ArrayOfHighlightTermsType"
minOccurs="0"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="FindItemResponseType">
        <xs:complexContent>
```

```
            <xs:extension base="m:BaseResponseMessageType"/>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="FindItemResponse" type="m:FindItemResponseType"/>
    <xs:complexType name="QueryStringType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="ResetCache" type="xs:boolean" use="optional"/>
                <xs:attribute name="ReturnHighlightTerms" type="xs:boolean" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```

7.2 Types Schema

Note *Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.*

This section contains the contents of the MS-OXWSSRCH-types.xsd file and information about additional files that this schema file requires to operate correctly.

MS-OXWSSRCH-types.xsd includes the file listed in the following table. For the schema file to operate correctly, this file has to be present in the folder that contains the WSDL, types schema, and messages schema files for this protocol.

File name	Defining specification
MS-OXWSCDATA-types.xsd	[MS-OXWSCDATA] section 7.2

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
elementFormDefault="qualified" version="Exchange2012" id="types">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
  <xs:complexType name="AggregateOnType">
    <xs:choice>
      <xs:element name="FieldURI" type="t:PathToUnindexedFieldType"/>
      <xs:element name="IndexedFieldURI" type="t:PathToIndexedFieldType"/>
      <xs:element name="ExtendedFieldURI" type="t:PathToExtendedFieldType"/>
    </xs:choice>
    <xs:attribute name="Aggregate" type="t:AggregateType" use="required"/>
  </xs:complexType>
  <xs:simpleType name="AggregateType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Minimum"/>
      <xs:enumeration value="Maximum"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="AndType">
    <xs:complexContent>
      <xs:extension base="t:MultipleOperandBooleanExpressionType"/>
    </xs:complexContent>
```

```
  </xs:complexType>
  <xs:element name="And" type="t:AndType" substitutionGroup="t:SearchExpression"/>
  <xs:complexType name="ArrayOfGroupedItemsType">
    <xs:choice>
      <xs:element name="GroupedItems" type="t:GroupedItemsType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="BaseGroupByType" abstract="true">
    <xs:attribute name="Order" type="t:SortDirectionType" use="required"/>
  </xs:complexType>
  <xs:complexType name="BasePagingType" abstract="true">
    <xs:attribute name="MaxEntriesReturned" type="xs:int" use="optional"/>
  </xs:complexType>
  <xs:simpleType name="ContainmentModeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="FullString"/>
      <xs:enumeration value="Prefixed"/>
      <xs:enumeration value="Substring"/>
      <xs:enumeration value="PrefixOnWords"/>
      <xs:enumeration value="ExactPhrase"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ContainmentComparisonType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Exact"/>
      <xs:enumeration value="IgnoreCase"/>
      <xs:enumeration value="IgnoreNonSpacingCharacters"/>
      <xs:enumeration value="Loose"/>
      <xs:enumeration value="IgnoreCaseAndNonSpacingCharacters"/>
      <xs:enumeration value="LooseAndIgnoreCase"/>
      <xs:enumeration value="LooseAndIgnoreNonSpace"/>
      <xs:enumeration value="LooseAndIgnoreCaseAndIgnoreNonSpace"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="ContainsExpressionType">
    <xs:complexContent>
      <xs:extension base="t:SearchExpressionType">
        <xs:sequence>
          <xs:element ref="t:Path"/>
          <xs:element name="Constant" type="t:ConstantValueType"/>
        </xs:sequence>
        <xs:attribute name="ContainmentMode" type="t:ContainmentModeType" use="optional"/>
        <xs:attribute name="ContainmentComparison" type="t:ContainmentComparisonType"
use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="Contains" type="t:ContainsExpressionType"
substitutionGroup="t:SearchExpression"/>
  <xs:complexType name="DistinguishedGroupByType">
    <xs:complexContent>
      <xs:extension base="t:BaseGroupByType">
        <xs:sequence>
          <xs:element name="StandardGroupBy" type="t:StandardGroupByType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```
<xs:complexType name="ExcludesValueType">
  <xs:attribute name="Value" type="t:ExcludesAttributeType" use="required"/>
</xs:complexType>
<xs:complexType name="ExcludesType">
  <xs:complexContent>
    <xs:extension base="t:SearchExpressionType">
      <xs:sequence>
        <xs:element ref="t:Path"/>
        <xs:element name="Bitmask" type="t:ExcludesValueType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="Excludes" type="t:ExcludesType" substitutionGroup="t:SearchExpression"/>
<xs:complexType name="ExistsType">
  <xs:complexContent>
    <xs:extension base="t:SearchExpressionType">
      <xs:sequence>
        <xs:element ref="t:Path"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="Exists" type="t:ExistsType" substitutionGroup="t:SearchExpression"/>
<xs:complexType name="FieldOrderType">
  <xs:sequence>
    <xs:element ref="t:Path"/>
  </xs:sequence>
  <xs:attribute name="Order" type="t:SortDirectionType" use="required"/>
</xs:complexType>
<xs:complexType name="FindFolderParentType">
  <xs:sequence>
    <xs:element name="Folders" type="t:ArrayOfFoldersType" minOccurs="0"/>
  </xs:sequence>
  <xs:attributeGroup ref="t:FindResponsePagingAttributes"/>
</xs:complexType>
<xs:complexType name="FindItemParentType">
  <xs:choice>
    <xs:element name="Items" type="t:ArrayOfRealItemsType"/>
    <xs:element name="Groups" type="t:ArrayOfGroupedItemsType"/>
  </xs:choice>
  <xs:attributeGroup ref="t:FindResponsePagingAttributes"/>
</xs:complexType>
<xs:simpleType name="FolderQueryTraversalType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Shallow"/>
    <xs:enumeration value="Deep"/>
    <xs:enumeration value="SoftDeleted"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="FractionalPageViewType">
  <xs:complexContent>
    <xs:extension base="t:BasePagingType">
      <xs:attribute name="Numerator" type="xs:int" use="required"/>
      <xs:attribute name="Denominator" type="xs:int" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="GroupByType">
```

```
      <xs:complexContent>
        <xs:extension base="t:BaseGroupByType">
          <xs:sequence>
            <xs:choice>
              <xs:element name="FieldURI" type="t:PathToUnindexedFieldType"/>
              <xs:element name="IndexedFieldURI" type="t:PathToIndexedFieldType"/>
              <xs:element name="ExtendedFieldURI" type="t:PathToExtendedFieldType"/>
            </xs:choice>
            <xs:element name="AggregateOn" type="t:AggregateOnType"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="GroupedItemsType">
      <xs:sequence>
        <xs:element name="GroupIndex" type="xs:string"/>
        <xs:element name="Items" type="t:ArrayOfRealItemsType"/>
      </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="IndexBasePointType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="Beginning"/>
        <xs:enumeration value="End"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="IndexedPageViewType">
      <xs:complexContent>
        <xs:extension base="t:BasePagingType">
          <xs:attribute name="Offset" type="xs:int" use="required"/>
          <xs:attribute name="BasePoint" type="t:IndexBasePointType" use="required"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="IsEqualToType">
      <xs:complexContent>
        <xs:extension base="t:TwoOperandExpressionType"/>
      </xs:complexContent>
    </xs:complexType>
    <xs:element name="IsEqualTo" type="t:IsEqualToType"
  substitutionGroup="t:SearchExpression"/>
    <xs:complexType name="IsNotEqualToType">
      <xs:complexContent>
        <xs:extension base="t:TwoOperandExpressionType"/>
      </xs:complexContent>
    </xs:complexType>
    <xs:element name="IsNotEqualTo" type="t:IsNotEqualToType"
  substitutionGroup="t:SearchExpression"/>
    <xs:complexType name="IsGreaterThanType">
      <xs:complexContent>
        <xs:extension base="t:TwoOperandExpressionType"/>
      </xs:complexContent>
    </xs:complexType>
    <xs:element name="IsGreaterThan" type="t:IsGreaterThanType"
  substitutionGroup="t:SearchExpression"/>
    <xs:complexType name="IsGreaterThanOrEqualToType">
      <xs:complexContent>
        <xs:extension base="t:TwoOperandExpressionType"/>
      </xs:complexContent>
    </xs:complexType>
```

```
  <xs:element name="IsGreaterThanOrEqualTo" type="t:IsGreaterThanOrEqualToType"
substitutionGroup="t:SearchExpression"/>
  <xs:complexType name="IsLessThanType">
    <xs:complexContent>
      <xs:extension base="t:TwoOperandExpressionType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="IsLessThan" type="t:IsLessThanType"
substitutionGroup="t:SearchExpression"/>
  <xs:complexType name="IsLessThanOrEqualToType">
    <xs:complexContent>
      <xs:extension base="t:TwoOperandExpressionType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="IsLessThanOrEqualTo" type="t:IsLessThanOrEqualToType"
substitutionGroup="t:SearchExpression"/>
  <xs:simpleType name="ItemQueryTraversalType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Shallow"/>
      <xs:enumeration value="SoftDeleted"/>
      <xs:enumeration value="Associated"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="NotType">
    <xs:complexContent>
      <xs:extension base="t:SearchExpressionType">
        <xs:sequence>
          <xs:element ref="t:SearchExpression"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="Not" type="t:NotType" substitutionGroup="t:SearchExpression"/>
  <xs:complexType name="MultipleOperandBooleanExpressionType" abstract="true">
    <xs:complexContent>
      <xs:extension base="t:SearchExpressionType">
        <xs:sequence>
          <xs:element ref="t:SearchExpression" minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="OrType">
    <xs:complexContent>
      <xs:extension base="t:MultipleOperandBooleanExpressionType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="Or" type="t:OrType" substitutionGroup="t:SearchExpression"/>
  <xs:complexType name="NonEmptyArrayOfFieldOrdersType">
    <xs:sequence>
      <xs:element name="FieldOrder" type="t:FieldOrderType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RestrictionType">
    <xs:sequence>
      <xs:element ref="t:SearchExpression"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="SearchExpressionType" abstract="true"/>
```

```
<xs:element name="SearchExpression" type="t:SearchExpressionType"/>
<xs:simpleType name="SearchFolderTraversalType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Shallow"/>
    <xs:enumeration value="Deep"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="SearchFolderType">
  <xs:complexContent>
    <xs:extension base="t:FolderType">
      <xs:sequence>
        <xs:element name="SearchParameters" type="t:SearchParametersType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SearchParametersType">
  <xs:sequence>
    <xs:element name="Restriction" type="t:RestrictionType"/>
    <xs:element name="BaseFolderIds" type="t:NonEmptyArrayOfBaseFolderIdsType"/>
  </xs:sequence>
  <xs:attribute name="Traversal" type="t:SearchFolderTraversalType" use="optional"/>
</xs:complexType>
<xs:complexType name="SeekToConditionPageViewType">
  <xs:complexContent>
    <xs:extension base="t:BasePagingType">
      <xs:sequence>
        <xs:element name="Condition" type="t:RestrictionType" minOccurs="1"/>
      </xs:sequence>
      <xs:attribute name="BasePoint" type="t:IndexBasePointType" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="SortDirectionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Ascending"/>
    <xs:enumeration value="Descending"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="StandardGroupByType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ConversationTopic"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="TwoOperandExpressionType" abstract="true">
  <xs:complexContent>
    <xs:extension base="t:SearchExpressionType">
      <xs:sequence>
        <xs:element ref="t:Path"/>
        <xs:element name="FieldURIOrConstant" type="t:FieldURIOrConstantType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs: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® Exchange Server 2007
- Microsoft® Exchange Server 2010
- Microsoft® Exchange Server 2013 Preview
- Microsoft® Lync™ 2010
- Microsoft® Lync® 2013 Preview

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: Exchange 2007 and Exchange 2010 do not implement the **SeekToConditionPageViewType** complex type.

<2> Section 2.2.4.28: Exchange 2007 and Exchange 2010 do not implement the **SeekToConditionPageViewType**.

<3> Section 3.1.4.1.1.1: Exchange 2007 and Exchange 2010 do not use the **ManagementRole** part.

<4> Section 3.1.4.2: Exchange 2007, Exchange 2010, and Exchange 2010 SP1 do not include the **DateTimePrecision** part.

<5> Section 3.1.4.2.1.1: Exchange 2007, the initial release version of Exchange 2010, and Exchange 2010 SP1 do not use the **DateTimePrecision** part.

<6> Section 3.1.4.2.1.1: Exchange 2007 and Exchange 2010 do not use the **ManagementRole** part.

<7> Section 3.1.4.2.3.2: Exchange 2007 and the initial release version of Exchange 2010 return all properties defined for any item in the specified folder.

9 Change Tracking

This section identifies changes that were made to the [MS-OXWSSRCH] protocol document between the April 2012 and July 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
6 Appendix A: Full WSDL	Updated WSDL.	N	Content updated.
7.1 Messages Schema	Updated messages schema.	N	Content updated.
7.1 Messages Schema	Updated messages schema.	N	Content updated.
7.2 Types Schema	Updated types schema.	N	Content updated.
8 Appendix C: Product Behavior	Added Lync 2010 and Lync 2013 Preview to the list of products.	Y	Content updated.

10 Index

A

Abstract data model
  server 36
Applicability 9
Attribute groups 35
Attributes 35

C

Capability negotiation 9
Change tracking 68
Complex types 14
  m:FindFolderResponseMessageType Complex Type 16
  m:FindItemResponseMessageType Complex Type 16
  t:AndType Complex Type 17
  t:ArrayOfGroupedItemsType Complex Type 17
  t:BasePagingType Complex Type 18
  t:ContainsExpressionType Complex Type 18
  t:ExcludesType Complex Type 19
  t:ExcludesValueType Complex Type 20
  t:ExistsType Complex Type 20
  t:FindFolderParentType Complex Type 21
  t:FindItemParentType Complex Type 21
  t:FractionalPageViewType Complex Type 22
  t:GroupedItemsType Complex Type 23
  t:IndexedPageViewType Complex Type 23
  t:IsEqualToType Complex Type 24
  t:IsGreaterThanOrEqualToType Complex Type 24
  t:IsGreaterThanType Complex Type 25
  t:IsLessThanOrEqualToType Complex Type 25
  t:IsLessThanType Complex Type 25
  t:IsNotEqualToType Complex Type 26
  t:MultipleOperandBooleanExpressionType Complex Type 26
  t:NotType Complex Type 27
  t:OrType Complex Type 27
  t:RestrictionType Complex Type 27
  t:SearchExpressionType Complex Type 28
  t:SearchFolderType Complex Type 28
  t:SearchParametersType Complex Type 28
  t:SeekToConditionPageViewType Complex Type 29
  t:TwoOperandExpressionType Complex Type 30

D

Data model - abstract
  server 36

E

Elements
  tns:And Element 11
  tns:Contains Element 12
  tns:Excludes Element 12
  tns:Exists Element 12
  tns:IsEqualTo Element 12
  tns:IsGreaterThan Element 12
  tns:IsGreaterThanOrEqualTo Element 13
  tns:IsLessThan Element 13
  tns:IsLessThanOrEqualTo Element 13
  tns:IsNotEqualTo Element 13
  tns:Not Element 13
  tns:Or Element 14
  tns:SearchExpression Element 14
Events
  local - server 53
  timer - server 53

F

Fields - vendor-extensible 9
Full WSDL 56
Full XML Schema 59
  Messages Schema 59
  Types Schema 61

G

Glossary 6
Groups 35

I

Implementer - security considerations 55
Index of security parameters 55
Informative references 7
Initialization
  server 36
Introduction 6

L

Local events
  server 53

M

m:FindFolderResponseMessageType Complex Type
  complex type 16
m:FindItemResponseMessageType Complex Type
  complex type 16
Message processing
  server 36
Messages
  attribute groups 35
  attributes 35
  complex types 14
  elements 10
  enumerated 10
  groups 35
  m:FindFolderResponseMessageType Complex Type complex type 16

m:FindItemResponseMessageType Complex Type complex type 16
namespaces 10
simple types 31
syntax 10
t:AndType Complex Type complex type 17
t:ArrayOfGroupedItemsType Complex Type complex type 17
t:BasePagingType Complex Type complex type 18
t:ContainmentComparisonType Simple Type simple type 31
t:ContainmentModeType Simple Type simple type 32
t:ContainsExpressionType Complex Type complex type 18
t:ExcludesType Complex Type complex type 19
t:ExcludesValueType Complex Type complex type 20
t:ExistsType Complex Type complex type 20
t:FindFolderParentType Complex Type complex type 21
t:FindItemParentType Complex Type complex type 21
t:FolderQueryTraversalType Simple Type simple type 33
t:FractionalPageViewType Complex Type complex type 22
t:GroupedItemsType Complex Type complex type 23
t:IndexBasePointType Simple Type simple type 34
t:IndexedPageViewType Complex Type complex type 23
t:IsEqualToType Complex Type complex type 24
t:IsGreaterThanOrEqualToType Complex Type complex type 24
t:IsGreaterThanType Complex Type complex type 25
t:IsLessThanOrEqualToType Complex Type complex type 25
t:IsLessThanType Complex Type complex type 25
t:IsNotEqualToType Complex Type complex type 26
t:ItemQueryTraversalType Simple Type simple type 34
t:MultipleOperandBooleanExpressionType Complex Type complex type 26
t:NotType Complex Type complex type 27
t:OrType Complex Type complex type 27
t:RestrictionType Complex Type complex type 27
t:SearchExpressionType Complex Type complex type 28
t:SearchFolderTraversalType Simple Type simple type 35
t:SearchFolderType Complex Type complex type 28
t:SearchParametersType Complex Type complex type 28
t:SeekToConditionPageViewType Complex Type complex type 29
t:TwoOperandExpressionType Complex Type complex type 30
tns:And Element element 11
tns:Contains Element element 12
tns:Excludes Element element 12
tns:Exists Element element 12
tns:IsEqualTo Element element 12
tns:IsGreaterThan Element element 12
tns:IsGreaterThanOrEqualTo Element element 13
tns:IsLessThan Element element 13
tns:IsLessThanOrEqualTo Element element 13
tns:IsNotEqualTo Element element 13
tns:Not Element element 13
tns:Or Element element 14
tns:SearchExpression Element element 14
transport 10

N

Namespaces 10
Normative references 6

O

Operations
FindFolder Operation 36
FindItem Operation 41
Overview (synopsis) 8

P

Parameters - security index 55
Preconditions 9
Prerequisites 9
Product behavior 67

R

References 6
informative 7
normative 6
Relationship to other protocols 8

S

Security
implementer considerations 55
parameter index 55
Sequencing rules
server 36
Server
abstract data model 36
FindFolder Operation operation 36
FindItem Operation operation 41
initialization 36
local events 53
message processing 36
sequencing rules 36
timer events 53
timers 36
Simple types 31
t:ContainmentComparisonType Simple Type 31

t:ContainmentModeType Simple Type 32
t:FolderQueryTraversalType Simple Type 33
t:IndexBasePointType Simple Type 34
t:ItemQueryTraversalType Simple Type 34
t:SearchFolderTraversalType Simple Type 35
Standards assignments 9
Syntax
messages - overview 10

T

t:AndType Complex Type complex type 17
t:ArrayOfGroupedItemsType Complex Type complex type 17
t:BasePagingType Complex Type complex type 18
t:ContainmentComparisonType Simple Type simple type 31
t:ContainmentModeType Simple Type simple type 32
t:ContainsExpressionType Complex Type complex type 18
t:ExcludesType Complex Type complex type 19
t:ExcludesValueType Complex Type complex type 20
t:ExistsType Complex Type complex type 20
t:FindFolderParentType Complex Type complex type 21
t:FindItemParentType Complex Type complex type 21
t:FolderQueryTraversalType Simple Type simple type 33
t:FractionalPageViewType Complex Type complex type 22
t:GroupedItemsType Complex Type complex type 23
t:IndexBasePointType Simple Type simple type 34
t:IndexedPageViewType Complex Type complex type 23
t:IsEqualToType Complex Type complex type 24
t:IsGreaterThanOrEqualToType Complex Type complex type 24
t:IsGreaterThanType Complex Type complex type 25
t:IsLessThanOrEqualToType Complex Type complex type 25
t:IsLessThanType Complex Type complex type 25
t:IsNotEqualToType Complex Type complex type 26
t:ItemQueryTraversalType Simple Type simple type 34
t:MultipleOperandBooleanExpressionType Complex Type complex type 26
t:NotType Complex Type complex type 27
t:OrType Complex Type complex type 27
t:RestrictionType Complex Type complex type 27
t:SearchExpressionType Complex Type complex type 28
t:SearchFolderTraversalType Simple Type simple type 35
t:SearchFolderType Complex Type complex type 28
t:SearchParametersType Complex Type complex type 28
t:SeekToConditionPageViewType Complex Type complex type 29
t:TwoOperandExpressionType Complex Type complex type 30
Timer events
server 53
Timers
server 36
tns:And Element element 11
tns:Contains Element element 12
tns:Excludes Element element 12
tns:Exists Element element 12
tns:IsEqualTo Element element 12
tns:IsGreaterThan Element element 12
tns:IsGreaterThanOrEqualTo Element element 13
tns:IsLessThan Element element 13
tns:IsLessThanOrEqualTo Element element 13
tns:IsNotEqualTo Element element 13
tns:Not Element element 13
tns:Or Element element 14
tns:SearchExpression Element element 14
Tracking changes 68
Transport 10
Types
complex 14
simple 31

V

Vendor-extensible fields 9
Versioning 9

W

WSDL 56

X

XML Schema 59
Messages Schema 59
Types Schema 61