[MS-OXWSSRCH]:
Mailbox Search Web Service Protocol Specification

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.

Table of Contents

1 Introduction

This document specifies the Mailbox Search Web Service protocol, which searches the contents of a **mailbox** and returns the specified **folders** or items. Clients use the SOAP protocol [SOAP1.1] to contact the mailbox search service.

1.1 Glossary

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

folder
Hypertext Transport Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
mailbox
property
search folder
SOAP body
SOAP fault
SOAP header
SOAP message
Uniform Resource Identifier (URI)
Web Services Description Language (WSDL)
WSDL message
WSDL port type
XML
XML namespace
XML schema

The following terms are specific to this document:

search folder: A folder that has dynamic contents that are specified by a set of search criteria.

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

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", June 2008.

[MS-OXWSCDATA] Microsoft Corporation, "Common Web Service Data Types", July 2009.

[MS-OXWSCONT] Microsoft Corporation, "Contacts Web Service Protocol Specification", July 2009.

[MS-OXWSFOLD] Microsoft Corporation, "Folders and Folder Permissions Web Service Protocol Specification", July 2009.

[MS-OXWSMSG] Microsoft Corporation, "E-Mail Message Types Web Service Protocol Specification", July 2009.

[MS-OXWSMTGS] Microsoft Corporation, "Calendaring Web Service Protocol Specification", July 2009.

[MS-OXWSPOST] Microsoft Corporation, "Post Items Web Service Protocol Specification", July 2009.

[MS-OXWSTASK] Microsoft Corporation, "Tasks Protocol Specification", July 2009.

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

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, http://www.ietf.org/rfc/rfc2396.txt.

[RFC2616] Fielding, R., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.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 Languages", RFC 3066, January 2001, http://www.ietf.org/rfc/rfc3066.txt.

[SOAP1.1] Box, D., 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] World Wide Web Consortium, "Namespaces in XML 1.0 (Second Edition)", August 2006, http://www.w3.org/TR/REC-xml-names/.

[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

None.

1.3 Protocol Overview

The Mailbox Search Web Service protocol provides clients with operations that enable them to search the contents of a mailbox on a server and to return the results of that search.

1.4 Relationship to Other Protocols

The Mailbox Search Web Service protocol uses SOAP over **HTTP**, as shown in the following figure.

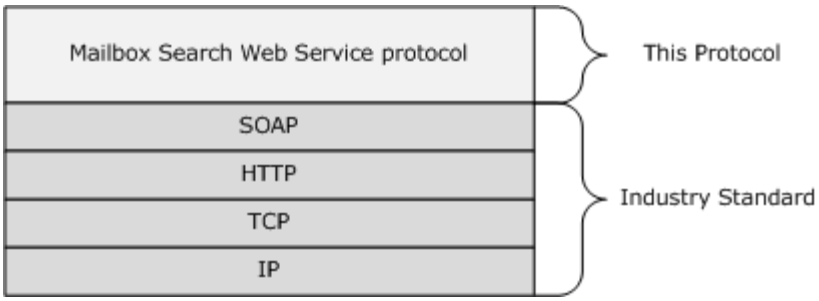

Figure 1: Mailbox Search Web Service protocol HTTP stack

The Mailbox Search Web Service protocol uses SOAP over **HTTPS**, as shown in the following figure.

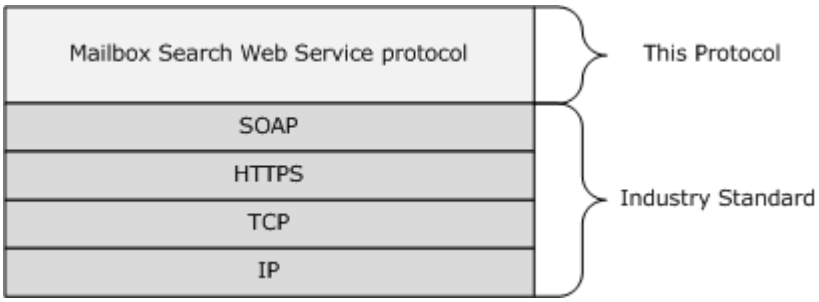

Figure 2: Mailbox Search Web Service protocol HTTPS stack

The Mailbox Search Web Service protocol specifies searches that identify items in the mailbox data store. After the item identifier is returned, one of the following protocols is used to return the information from the data store:

- Folders and Folder Permissions Web Service protocol [MS-OXWSFOLD]
- E-Mail Message Types Web Service protocol [MS-OXWSMSG]
- Calendaring Web Service protocol [MS-OXWSMTGS]
- Post Items Web Service protocol [MS-OXWSPOST]
- Tasks Web Service protocol [MS-OXWSTASK]

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

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

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses SOAP 1.1, as specified in section 2.1.
- **Protocol Versions:** This protocol specifies only one **WSDL port type** version.

- **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: None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

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

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	[MS-OXWSSRCH]
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
targetNamespace	http://schemas.microsoft.com/exchange/services/2006/messages	[MS-OXWSSRCH]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	[MS-OXWSSRCH]

2.2.2 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	Description
t:ContainmentComparisonType	Specifies whether a search is exact or whether it ignores casing and spaces.
t:ContainmentModeType	Specifies the search boundaries.
t:FolderQueryTraversalType	Specifies the types of subtree traversals for deletion and enumeration.
t: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.
t:ItemQueryTraversalType	Specifies whether a search finds items in folders or in the dumpster folder.

Simple Type	Description
t:SearchFolderTraversalType	Specifies the options for how a folder hierarchy is searched when the contents of a search folder are identified.

2.2.2.1 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>
```

Enumeration

The following values are defined by the **ContainmentComparisonType** simple type:

Value	Description
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.

Value	Description
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.

2.2.2.2 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>
```

Enumeration

The following values are defined by the **ContainmentModeType** simple type:

Value	Description
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**.

Value	Description
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**.

2.2.2.3 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>
```

Enumeration

The following values are defined by the **FolderQueryTraversalType** simple type:

Value	Description
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.
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.

2.2.2.4 t:IndexBasePointType Simple Type

The IndexBasePointType simple type specifies the whether a page of items that are returned by the FindFolder operation (section 3.1.4.1) or FindItem (section 3.1.4.2) operation 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>
```

Enumeration

The following values are defined by the **IndexBasePointType** simple type:

Value	Description
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.

2.2.2.5 t:ItemQueryTraversalType Simple Type

The ItemQueryTraversalType simple type specifies whether the search finds items in folders or in the dumpster folder.

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

Enumeration

The following values 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 dumpster folder are returned.

2.2.2.6 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>
```

Enumeration

The following values are defined by the **SearchFolderTraversalType** simple type:

Value	Description
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.

2.2.3 Complex Types

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	Description
m:FindFolderResponseMessageType	Specifies the response message for the FindFolder operation (section 3.1.4.1).
m:FindItemResponseMessageType	Specifies the result body from the FindItem operation (section 3.1.4.2).
t:AndType	Specifies a search expression that performs a Boolean **AND** operation between two or more search expressions.
t:ArrayOfGroupedItemsType	Specifies an array of items that are returned by the FindItem operation (section 3.1.4.2).
t:BasePagingType	Specifies the base type for derived types that specify paged views.
t:ContainsExpressionType	Specifies a search expression that determines whether a given property contains the supplied constant string value.
t:ExcludesType	Specifies a bitwise mask of a property for an exclude search restriction.
t:ExcludesValueType	Specifies a hexadecimal or decimal mask for an Excludes restriction.
t:ExistsType	Specifies a search restriction that resolves to true if the supplied property exists on an item.
t:FindFolderParentType	Specifies the results of searching a single root folder.
t:FindItemParentType	Specifies the results of searching a single root folder.

Complex Type	Description
t:FractionalPageViewType	Specifies where a paged view starts and the maximum number of items that are returned.
t:GroupedItemsType	Specifies a collection of items that are the result of a grouped FindItem operation (section 3.1.4.2).
t:IndexedPageViewType	Specifies how paged item information is returned by the FindItem operation (section 3.1.4.2) operation or the FindFolder operation (section 3.1.4.1).
t: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.
t: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.
t: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.
t: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.
t: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.
t: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.
t:MultipleOperandBooleanExpressionType	Specifies the base type for search expressions that are formed by two or more Boolean operands.
t:NotType	Specifies a search expression that negates the Boolean value of the search expression that it contains.
t:OrType	Specifies a search expression that performs a logical **OR** on the search expression that it contains.
t:RestrictionType	Specifies a search restriction or query for the FindItem operation (section 3.1.4.2) operation or the FindFolder operation (section 3.1.4.1).
t:SearchExpressionType	Specifies the base type for all search expressions.
t:SearchFolderType	Specifies a representation of a **search folder** that is contained in a mailbox.
t:SearchParametersType	Specifies the search parameters that define the contents of a search folder.
t:TwoOperandExpressionType	Specifies the base type for derived classes that represent a restriction that is formed by comparing two values against one

Complex Type	Description
	another.

2.2.3.1 m:FindFolderResponseMessageType Complex Type

The FindFolderResponseMessageType complex type specifies the response message for the FindFolder operation (section 3.1.4.1). The FindFolderResponseMessageType complex type extends the ResponseMessageType complex type ([MS-OXWSCDATA] section 2.2.3.49).

```
<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>
```

Child Elements

Element	Type	Description
RootFolder	t:FindFolderParentType	Specifies an array of folders and paging information that is returned by the FindFolder operation (section 3.1.4.1).

2.2.3.2 m:FindItemResponseMessageType Complex Type

The FindItemResponseMessageType complex type specifies the result body from the FindItem operation (section 3.1.4.2). The FindItemResponseMessageType complex type extends the m:ResponseMessageType complex type ([MS-OXWSCDATA] section 2.2.3.49).

```
<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>
```

Child Elements

Element	Type	Description
RootFolder	t:FindFolderParentType	Specifies the root folder of the reponse.

2.2.3.3 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.

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

2.2.3.4 t:ArrayOfGroupedItemsType Complex Type

The ArrayOfGroupedItemsType complex type specifies an array of items that are returned by the FindItem operation (section 3.1.4.2).

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

Child Elements

Element	Type	Description
GroupedItems	t:GroupedItemsType	Specifies an array of items that are returned by the FindItem operation (section 3.1.4.2).

2.2.3.5 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>
```

Attributes

Name	Type	Description
MaxEntriesReturned	xs:int	Specifies the maximum number of entries that are returned with each page of the response. This attribute can be specified.

2.2.3.6 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>
```

Child Elements

Element	Type	Description
t:Path	t:Path	Specifies the property to use in a contains search expression. The t:Path element ([MSOXWSCDATA] section 2.2.4.6) specifies a **substitutionGroup**, as specified in [XMLSCHEMA0]. An element that is represented by the <Path> **substitutionGroup** MUST be present.
Constant	t:ConstantValueType	Specifies a constant value for a search restriction.

Attributes

Name	Type	Description
ContainmentMode	t:ContainmentModeType	Specifies the boundaries of a search.
ContainmentComparison	t:ContainmentComparisonType	Specifies whether a search ignores cases and spaces.

2.2.3.7 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>
```

Child Elements

Element	Type	Description
t:Path	t:Path	Specifies the property to use in an Excludes search expression. The t:Path element ([MS-OXWSCDATA] section 2.2.4.6) specifies a **substitutionGroup**, as specified in [XMLSCHEMA0]. An element that is represented by the <Path> **substitutionGroup** MUST be present.
Bitmask	t:ExcludesValueType	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.

2.2.3.8 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>
```

Attributes

Name	Type	Description
Value	t:ExcludesAttributeType	Specifies a decimal or hexadecimal bitmask for an exclude restriction. This attribute MUST be present for an Excludes restriction.

2.2.3.9 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 (section 2.2.3.25).

```
<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>
```

Child Elements

Element	Type	Description
t:Path	t:Path	Specifies the property to use in an Excludes search expression. The t:Path element ([MS-OXWSCDATA] section 2.2.4.6) specifies a **substitutionGroup**, as specified in [XMLSCHEMA0]. An element that is represented by the <Path> **substitutionGroup** MUST be present.

2.2.3.10 t:FindFolderParentType Complex Type

__Description__

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

Child Elements

Element	Type	Description
Folders	t:ArrayOfFoldersType	Specifies the folders in the result set.

Attribute Groups

Name
t:FindResponsePagingAttributes

2.2.3.11 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>
```

Child Elements

Element	Type	Description
Items	t:ArrayOfRealItemsType	Specifies the results of a search in which the items returned are not grouped.
Groups	t:ArrayOfGroupedItemsType	Specifies the grouped results of a search.

Attribute Groups

Name
t:FindResponsePagingAttributes

If a <GroupBy> or <DistinguishedGroupBy> element was specified in the FindItemType complex type (section 3.1.4.2.2.2) instance that was sent to the FindItem operation (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.

2.2.3.12 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 (section 2.2.3.5).

```
<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>
```

Attributes

Name	Type	Description
Denominator	xs:int	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.

2.2.3.13 t:GroupedItemsType Complex Type

The GroupedItemsType complex type specifies a collection of items that are the result of a grouped FindItem operation (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>
```

Child Elements

Element	Type	Description
GroupIndex	xs:string	Specifies the property value that is used to group the items.
Items	t:ArrayOfRealItemsType	Specifies the group of items that correspond to the specified group value.

2.2.3.14 t:IndexedPageViewType Complex Type

The IndexedPageViewType complex type specifies how paged item information is returned by the FindItem operation (section 3.1.4.2) or the FindFolder operation (section 3.1.4.1). The IndexedPageViewType complex type extends the BasePagingType complex type (section 2.2.3.5).

```
<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>
```

Attributes

Name	Type	Description
Offset	xs:ing	Specifies the offset from the <BasePoint> element. This attribute MUST be specified.
BasePoint	t:IndexBasePointType	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.

2.2.3.15 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 (section 2.2.3.28).

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

2.2.3.16 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 (section 2.2.3.28).

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

2.2.3.17 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 (section 2.2.3.28).

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

2.2.3.18 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 (section 2.2.3.28).

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

2.2.3.19 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 (section 2.2.3.28).

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

2.2.3.20 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 (section 2.2.3.28).

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

2.2.3.21 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 (section 2.2.3.25).

```
<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>
```

Child Elements

Element	Type	Description
t:SearchExpression	t:SearchExpression	Specifies an array of search expressions that represents a set of operands.

2.2.3.22 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 (section 2.2.3.25).

```
<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>
```

Child Elements

Element	Type	Description
t:SearchExpression	t:SearchExpression	Specifies a search expression.

2.2.3.23 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 (section 2.2.3.21).

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

2.2.3.24 t:RestrictionType Complex Type

The RestrictionType complex type specifies a search restriction or query for a FindItem operation (section 3.1.4.2) or a FindFolder operation (section 3.1.4.1).

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

Child Elements

Element	Type	Description
t:SearchExpression	t:SearchExpression	Specifies the search term that defines the restriction.

2.2.3.25 t:SearchExpressionType Complex Type

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

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

2.2.3.26 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>
```

Child Elements

Element	Type	Description
SearchParameters	t:SearchParametersType	Specifies the search parameters that define the contents of the search folder. This element can be present.

2.2.3.27 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>
```

Child Elements

Element	Type	Description
Restriction	t:RestrictionType	Specifies the query that defines the contents of a search folder.
BaseFolderIds	t:NonEmptyArrayOfBaseFolderIdsType	Specifies an array of folder identifiers that identify the folders that are searched.

Attributes

Name	Type	Description
Traversal	t:SearchFolderTraversalType	Specifies the depth of a search folder sub-tree traversal.

2.2.3.28 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 (section 2.2.3.25).

```
<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>
```

Child Elements

Element	Type	Description
t:Path	t:Path	Specifies the property path that is searched on for each item or folder in a search.
FieldURIOrConstant	t:FieldURIOrConstantType	Specifies the property or constant that is compared with each item or folder in a restriction.

2.2.4 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	Description
t:And	Specifies a search expression that allows you to perform a Boolean **AND** operation between two or more search expressions.
t:Contains	Specifies a search expression that determines whether a given property contains the supplied constant string value.
t:Excludes	Specifies a search expression that allows you to perform a bitwise mask of the specified property and a supplied value.
t:Exists	Specifies a search expression that determines whether the specified property exists on an item.
t: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.
t: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.
t: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.
t:IsLessThan	Specifies a search expression that compares a property to either a constant

Element	Description
	value or another property and evaluates to **true** if the first property is less than the second.
t: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.
t: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.
t:Not	Specifies a search expression that negates the Boolean value of a search expression that it contains.
t:Or	Specifies a search expression that performs a logical **OR** on the search expressions that it contains and returns **true** if any of the search expressions return **true**.
t:SearchExpression	Specifies the base schema type for all search expressions. This type is abstract and will never occur directly within instance documents.

2.2.4.1 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"
 />
```

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

2.2.4.2 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"
 />
```

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

2.2.4.3 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"
   />
```

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

2.2.4.4 Exists Element

The Exists element specifies a search expression 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"
 />
```

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

2.2.4.5 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"
 />
```

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

2.2.4.6 IsGreaterThan Element

The IsGreaterThan element 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.

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

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

2.2.4.7 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"
```

```
/>
```

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

2.2.4.8 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"
 />
```

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

2.2.4.9 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"
 />
```

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

2.2.4.10 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"
 />
```

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

2.2.4.11 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"
 />
```

2.2.4.12 Or Element

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

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

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

2.2.4.13 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"
 />
```

2.2.5 Attributes

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

2.2.6 Groups

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

2.2.7 Attribute Groups

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

2.2.8 Messages

This specification does not define any common **XML schema** message 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.

Operation	Description
FindFolder	Searches the data store for a specified folder.
FindItem	Searches the data store for a specified item.

3.1.1 Abstract Data Model

The Mailbox Search Web Service protocol is a stateless protocol.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

This protocol includes the two operations that are listed in the following table.

Operation	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

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

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

Request

Message Format	Description
tns:FindFolderSoapIn	Specifies the **SOAP message** that contains the operation parameters.

Response

Message Format	Description
tns:FindFolderSoapOut	Specifies the SOAP message that contains the search results.

3.1.4.1.1 Complex Types

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

3.1.4.1.1.1 m:FindFolderResponseType Complex Type

The FindFolderResponseType complex type extends the BaseResponseMessageType complex type ([MS-OXWSCDATA] section 2.2.3.15).

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

3.1.4.1.1.2 m:FindFolderType Complex Type

The **FindFolderType** complex type specifies a request to find folders in a mailbox. The **FindFolderType** complex type extends the **m:BaseRequestType** complex type ([MS-OXWSCDATA] section 2.2.3.14).

```
<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>
```

Child Elements

Element	Type	Description
FolderShape	t:FolderResponseShapeType	Specifies the contents of the query response.
IndexedPageFolderView	t:IndexedPageViewType	Specifies how paged information is returned by the query.
FractionalPageFolderView	t:FractionalPageViewType	Specifies the starting item and the number of items that are returned by a paged query.
Restriction	t:RestrictionType	Specifies the search parameters that define the folder query.
ParentFolderIds	t:NonEmptyArrayOfBaseFolderIdsType	Specifies the folders that the query searches.

Attributes

Name	Type	Description
Traversal	t:FolderQueryTraversalType	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.

3.1.4.1.2 Elements

The following **XML schema** element definitions are specific to this operation.

3.1.4.1.2.1 FindFolder Element

The FindFolder element specifies the base element for a FindFolder operation (section 3.1.4.1) request.

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

3.1.4.1.2.2 FindFolderResponse Element

The FindFolderResponse element specifies the response message for a FindFolder operation (section 3.1.4.1).

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

3.1.4.1.3 Messages

The following **WSDL message** definitions are specific to this operation.

3.1.4.1.3.1 tns:FindFolderSoapIn Message

The FindFolderSoapIn message contains five parts, as described in the following table.

Part Name	Element/Type	Description
request	tns:FindFolder	Specifies the request.
Impersonation	t:ExchangeImpersonation	Specifies the user whom the client application is impersonating.
MailboxCulture	t:MailboxCulture	Specifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	t:RequestServerVersion	Specifies the schema version for the FindFolder operation (section 3.1.4.1) request.
TimeZoneContext	t:TimeZoneContext	Specifies 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.

3.1.4.1.3.2 tns:FindFolderSoapOut Message

The FindFolderSoapOut message contains two parts, as described in the following table.

Part Name	Element/Type	Description
FindFolderResult	tns:FindFolderResponse	Specifies the response.
ServerVersion	t:ServerVersionInfo	Specifies the server version for the response.

3.1.4.2 FindItem Operation

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

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

Request

Message Format	Description
tns:FindItemSoapIn	Specifies the SOAP message that requests the find items operation.

Response

Message Format	Description
tns:FindItemSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.2.1 Simple Types

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

3.1.4.2.1.1 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 (section 3.1.4.2).

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

Enumeration

The following values are defined by the **AggregateType** simple type:

Value	Description
Maximum	Specifies that the groups are sorted starting with the maximum value for a specified

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

3.1.4.2.1.2 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>
```

Enumeration

The following values are defined by the **SortDirectionType** simple type:

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

3.1.4.2.1.3 t:StandardGroupByType Simple Type

The StandardGroupByType simple type specifies the standard grouping and aggregating mechanisms for a grouped response to the **FindItem** operation (section 3.1.4.2).

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

Enumeration

The following value is defined by the **StandardGroupByType** simple type:

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

3.1.4.2.2 Complex Types

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

3.1.4.2.2.1 m:FindItemResponseType Complex Type

The FindItemResponseType complex type extends the BaseResponseMessageType complex type ([MS-OXWSCDATA] section 2.2.3.15).

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

3.1.4.2.2.2 m:FindItemType Complex Type

The **FindItemType** complex type specifies the search criteria to use for the **FindItem** operation (section 3.1.4.2). The **FindItemType** complex type extends the **m:BaseRequestType** complex type ([MS-OXWSCDATA] section 2.2.3.14).

```
<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>
```

Child Elements

Element	Type	Description
ItemShape	t:ItemResponseShapeType	Specifies the array of items that are returned by the query.
IndexedPageItemView	t:IndexedPageViewType	Specifies how paged item information is returned in the response.
FractionalPageItemView	t:FractionalPageViewType	Specifies the starting item and number of items to return by the query.
CalendarView	t:CalendarViewType	Specifies the settings that are used to return calendar items as they appear in a calendar.
ContactsView	t:ContactsViewType	Specifies the settings that are used to return contact items based on their alphabetical display names.
GroupBy	t:GroupByType	Specifies the grouping for items that are returned by a query.

Element	Type	Description
DistinguishedGroupBy	t:DistinguishedGroupByType	Specifies a standard grouping.
Restriction	t:RestrictionType	Specifies a search restriction or query.
SortOrder	t:NonEmptyArrayOfFieldOrdersType	Specifies one or more **t:FieldOrderType** elements (section 3.1.4.2.2.6) that specify how the results should be sorted.
ParentFolderIds	t:NonEmptyArrayOfBaseFolderIdsType	Specifies one or more folders that are the root of the search.
QueryString	xs:string	Specifies the query that is used for the search.

Attributes

Name	Type	Description
Traversal	t:ItemQueryTraversalType	Specifies whether the search finds items in folders or in the dumpster folder.

The **m:FindItemType** complex type specifies the search criteria to find a set of items by using the **FindItem** operation (section 3.1.4.2) 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, should be grouped.

- <GroupBy>
- <DistinguishedGroupBy>

3.1.4.2.2.3 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>
```

Child Elements

Element	Type	Description
FieldURI	t:PathToUnindexedFieldType	Specifies a well-known data store property that is used to group the items.
IndexedFieldURI	t:PathToIndexedFieldType	Specifies an individual member of a dictionary that is used to group the items.
ExtendedFieldURI	t:PathToExtendedFieldType	Specifies an extended property that is used to group the items.

Attributes

Name	Type	Description
Aggregate	t:AggregateType	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.

3.1.4.2.2.4 t:BaseGroupByType Complex Type

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

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

Attributes

Name	Type	Description
Order	t:SortDirectionType	Specifies the sort order of the returned grouped items. This attribute MUST be specified.

3.1.4.2.2.5 t:DistinguishedGroupByType Complex Type

The **DistinguishedGroupByType** complex type specifies a standard grouping for the **FindItem** operation (section 3.1.4.2). The **DistinguishedGroupByType** complex type extends the **BaseGroupByType** complex type (section 3.1.4.2.2.4).

```
<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>
```

Child Elements

Element	Type	Description
StandardGroupBy	t:StandardGroupByType	Specifies one of the standard groupings for returned items.

3.1.4.2.2.6 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>
```

Child Elements

Element	Type	Description
t:Path	t:Path	Specifies the **URI** that describes the field by which the results are sorted.

Attributes

Name	Type	Description
Order	t:SortDirectionType	Specifies the direction of the sort. This attribute MUST be specified.

3.1.4.2.2.7 t:GroupByType Complex Type

The **GroupByType** complex type specifies the grouping for items that are returned by the **FindItem** operation (section 3.1.4.2). The **GroupByType** complex type extends the **BaseGroupByType** complex type (section 3.1.4.2.2.4).

```
<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>
```

Child Elements

Element	Type	Description
FieldURI	t:PathToUnindexedFieldType	Specifies the URI to an unindexed item property.
IndexedFieldURI	t:PathToIndexedFieldType	Specifies the URI to an indexed item property.
ExtendedFieldURI	t:PathToExtendedFieldType	Specifies an extended item property.
AggregateOn	t:AggregateOnType	Specifies the item property that is used to determine

Element	Type	Description
		the order of groups in a response.

3.1.4.2.2.8 t:NonEmptyArrayOfFieldOrdersType Complex Type

The **NonEmptyArrayOfFieldOrdersType** complex type specifies an array of t:FieldOrderType complex type (section 3.1.4.2.2.6) 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>
```

Child Elements

Element	Type	Description
FieldOrder	t:FieldOrderType	Specifies one or more t:FieldOrderType complex types (section 3.1.4.2.2.6).

3.1.4.2.3 Elements

The following **XML schema** element definitions are specific to this operation.

3.1.4.2.3.1 FindItem Element

The **FindItem** element specifies the base element for a **FindItem** operation (section 3.1.4.2).

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

3.1.4.2.3.2 FindItemResponse Element

The **FindItemResponse** element specifies the response message for the **FindItem** operation (section 3.1.4.2).

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

3.1.4.2.4 Messages

The following **WSDL message** definitions are specific to this operation.

3.1.4.2.4.1 tns:FindItemSoapIn Message

The FindItemSoapIn message contains five parts, as described in the following table.

Part Name	Element/Type	Description
request	tns:FindFolder	Specifies the request.
Impersonation	t:ExchangeImpersonation	Specifies the user that the client application is impersonating.
MailboxCulture	t:MailboxCulture	Specifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	t:RequestServerVersion	Specifies the schema version for the **FindFolder** operation (section 3.1.4.1) request.
TimeZoneContext	t:TimeZoneContext	Specifies 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.

3.1.4.2.4.2 tns:FindItemSoapOut Message

The FindItemSoapOut message contains two parts, as described in the following table.

Part Name	Element/Type	Description
FindItemResult	tns:FindItemResponse	Specifies the response.
ServerVersion	t:ServerVersionInfo	Specifies the server version for the response.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Client Details

None.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Message Processing Events and Sequencing Rules

None.

3.2.5 Timer Events

None.

3.2.6 Other Local Events

None.

4 Protocol Examples

None.

5 Security

5.1 Security Considerations for Implementers

The Mailbox Search Web Service protocol does not use any additional security mechanisms.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

The following table lists the **XML** files that are required to implement the functionality that is 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.	6.1
MS-OXWSSRCH-types.xsd	Contains the XML schema type definitions that are used in this protocol.	6.2
MS-OXWSSRCH-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	6.3

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.

6.1 WSDL

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="Exchange2010"
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" />
               <!-- Add global elements and types from messages.xsd -->
          </xs:schema>
          <xs:schema id="types" elementFormDefault="qualified" version="Exchange2010"
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"/>
               <!-- Add global elements and types from types.xsd -->
          </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: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: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: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:message>
    <wsdl:message name="FindFolderSoapOut">
        <wsdl:part name="FindFolderResult" element="tns:FindFolderResponse"/>
        <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
    </wsdl:message>

</wsdl:definitions>
```

6.2 Types Schema

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 6.2

```
<?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="Exchange2010" 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" 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: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>
```

6.3 Messages Schema

This section contains the contents of the MS-OXWSSRCH-messages.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 in the folder that contains the WSDL, types schema, and messages schema files for this protocol.

File name	Defining specification
MS-OXWSCDATA-messages.xsd	[MS-OXWSCDATA] section 6.3

```
<?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="Exchange2010" id="messages">
<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="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="xs:string" minOccurs="0"/>
</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: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:schema>
```

7 Appendix B: Product Behavior

The information in this specification is applicable to the following product versions. References to product versions include released service packs.

- Microsoft Exchange Server 2010

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified.

Unless otherwise specified, any statement of optional behavior in this specification 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 product does not follow the prescription.

8 Change Tracking

This section identifies changes made to [MS-OXWSSRCH] protocol documentation between July 2009 and November 2009 releases. Changes are classed as major, minor, or editorial.

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.
- A protocol is deprecated.
- The removal of a document from the documentation set.
- Changes made for template compliance.

Minor changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

Editorial changes apply to grammatical, formatting, and style issues.

No changes means that the document is identical to its last release.

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

- New content added.
- Content update.
- 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 always have the revision type "Editorially updated."

Some important terms used in revision 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.

Changes are listed in the following table. If you need further information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.3.11 t:FindItemParentType Complex Type	52744 Added attribute group to element syntax.	N	Content update.

9 Index