[MS-OXWSFOLD]:

Folders and Folder Permissions Web Service Protocol

Revision Summary

Date	Revision History	Revision Class	Comments
7/15/2009	1.0	Major	Initial Availability.
11/4/2009	1.0.1	Editorial	Revised and edited the technical content.
2/10/2010	2.0.0	Major	Updated and revised the technical content.
5/5/2010	3.0.0	Major	Updated and revised the technical content.
8/4/2010	4.0	Major	Significantly changed the technical content.
11/3/2010	5.0	Major	Significantly changed the technical content.
3/18/2011	6.0	Major	Significantly changed the technical content.
8/5/2011	6.1	Minor	Clarified the meaning of the technical content.
10/7/2011	6.1	No Change	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	7.0	Major	Significantly changed the technical content.
4/27/2012	7.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	7.1	Minor	Clarified the meaning of the technical content.
10/8/2012	8.0	Major	Significantly changed the technical content.
2/11/2013	8.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	8.1	Minor	Clarified the meaning of the technical content.
11/18/2013	8.1	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	8.1	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	8.2	Minor	Clarified the meaning of the technical content.
7/31/2014	8.3	Minor	Clarified the meaning of the technical content.
10/30/2014	8.4	Minor	Clarified the meaning of the technical content.
5/26/2015	9.0	Major	Significantly changed the technical content.
9/14/2015	10.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Folders and Folder Permissions Web Service Protocol is used by clients to manipulate and organize folders and to modify folder permissions, which enable users to access or to perform certain operations on the folder.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable ASCII characters, as described in [RFC4648].

Calendar folder: A Folder object that contains Calendar objects.

contact: A person, company, or other entity that is stored in a directory and is associated with one or more unique identifiers and attributes (2), such as an Internet message address or login name.

contact identifier: A universally unique identifier (UUID) that identifies a partner in the MSDTC Connection Manager: OleTx Transports Protocol. These UUIDs are frequently converted to and from string representations. This string representation must follow the format specified in [C706] Appendix A. In addition, the UUIDs must be compared, as specified in [C706] Appendix A.

Contacts folder: A Folder object that contains Contact objects.

endpoint: A communication port that is exposed by an application server for a specific shared service and to which messages can be addressed.

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

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol Secure (HTTPS): An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [SSL3] and [RFC5246].

item: A unit of content that can be indexed and searched by a search application.

Junk Email folder: A **special folder** that is the default location for Message objects that are determined to be junk email by a Junk Email rule.

mailbox: A **message store** that contains email, calendar items, and other Message objects for a single recipient.

managed folder: A Folder object that is created by an administrator and placed in a user's mailbox for messaging records management purposes. The retention and journaling of messages in managed folders are controlled by managed content settings that are applied to the Folder object.

message store: A unit of containment for a single hierarchy of Folder objects, such as a mailbox or public folders.

Passport Unique ID (PUID): A unique user name associated with a Microsoft Passport account.

permission: A rule that is associated with an object and that regulates which users can gain access to the object and in what manner. See also rights.

public folder: A Folder object that is stored in a location that is publicly available.

search folder: A Folder object that provides a means of querying for items that match certain criteria. The search folder includes the search folder definition message and the search folder container.

Sent Items folder: A **special folder** that is the default location for storing copies of Message objects after they are submitted or sent.

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

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

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

SOAP header: A mechanism for implementing extensions to a **SOAP message** in a decentralized manner without prior agreement between the communicating parties. See [SOAP1.2-1/2007] section 5.2 for more information.

SOAP message: An **XML** document consisting of a mandatory SOAP envelope, an optional **SOAP header**, and a mandatory **SOAP body**. See [SOAP1.2-1/2007] section 5 for more information.

special folder: One of a default set of Folder objects that can be used by an implementation to store and retrieve user data objects.

Tasks folder: A Folder object that contains Task objects.

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

web server: A server computer that hosts websites and responds to requests from applications.

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

WSDL message: An abstract, typed definition of the data that is communicated during a WSDL operation [WSDL]. Also, an element that describes the data being exchanged between web service providers and clients.

WSDL port type: A named set of logically-related, abstract **Web Services Description Language (WSDL)** operations and messages.

XML: The Extensible Markup Language, as described in [XML1.0].

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

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by **XML** itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

1.2 References

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

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

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

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

[MS-OXWSCORE] Microsoft Corporation, "Core Items Web Service Protocol".

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

[MS-OXWSMSG] Microsoft Corporation, "Email Message Types Web Service Protocol".

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

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

[MS-OXWSSRCH] Microsoft Corporation, "Mailbox Search Web Service Protocol".

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

[MS-OXWSURPT] Microsoft Corporation, "Retention Tag Web Service Protocol".

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

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

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

[RFC3066] Alvestrand, H., "Tags for the Identification of Languages", BCP 47, RFC 3066, January 2001, http://www.ietf.org/rfc/rfc3066.txt

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

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

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

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

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

1.2.2 Informative References

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

[MS-OXPROTO] Microsoft Corporation, "Exchange Server Protocols System Overview".

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

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

1.3 Overview

The Folders and Folder Permissions Web Service Protocol enables clients to create, copy, move, delete, get, or empty folders, and to modify folder **permissions** that are stored on the server. Clients can also use this protocol to locate the folders and **special folders** in a **mailbox**.

1.4 Relationship to Other Protocols

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

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

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

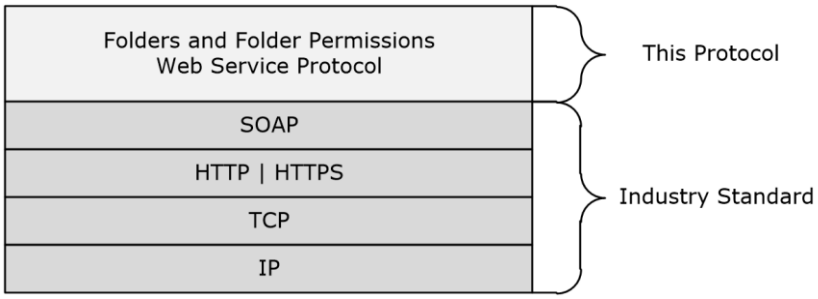

Figure 1: This protocol in relation to other protocols

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [MS-OXPROTO].

1.5 Prerequisites/Preconditions

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

1.6 Applicability Statement

This protocol is applicable to client applications that copy, create, delete, empty, get, move, or update folders. This Web service protocol is applicable to SOAP-based clients, as described in [SOAP1.1].

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with SOAP 1.1, as specified in section 2.1.
- **Protocol Versions:** This protocol has only one **WSDL port type** version. The **WSDL** version of the request is identified using the **RequestServerVersion** element, as described in [MS-OXWSCDATA] section 2.2.3.11 and the version of the server responding to the request is identified by using the **ServerVersionInfo** element, as described in [MS-OXWSCDATA] section 2.2.3.12.
- **Security and Authentication Methods:** This protocol relies on the web server that hosts the application to perform authentication.
- **Capability Negotiation:** This protocol does not support version negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

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

2.1 Transport

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

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

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema** as defined in [XMLSCHEMA1] and [XMLSCHEMA2], and Web Services Description Language (WSDL) as defined in [WSDL].

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
tns	http://schemas.microsoft.com/exchange/services/2006/messages	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
soap	http://schemas.xmlsoap.org/WSDL/soap/	[SOAP1.1]
WSDL	http://schemas.xmlsoap.org/WSDL/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA2]
t	http://schemas.microsoft.com/exchange/services/2006/types	
m	http://schemas.microsoft.com/exchange/services/2006/messages	

2.2.2 Messages

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

2.2.3 Elements

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

2.2.4 Complex Types

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

Complex type name	Description
AddressListIdType	Specifies an identifier for an address list.
ArrayOfFoldersType	Specifies an array of folders.
ArrayOfUnknownEntriesType	Represents an array of unknown permission entries that cannot be resolved against the directory service.
BaseFolderIdType	Specifies the base type for derived types that represent a folder identifier.
BaseFolderType	Specifies the base type for folders.
BaseMoveCopyFolderType	Specifies the base type for the **CopyFolderType** (section 3.1.4.1.3.2) and **MoveFolderType** (section 3.1.4.7.3.2) complex types.
BasePermissionType	Specifies the base type that defines base permissions for **items** and folders.
FolderChangeDescriptionType	Specifies a change to a single folder property.
FolderChangeType	Specifies a collection of changes to be performed on a single folder.
FolderInfoResponseMessageType	Represents the response message.
FolderType	Represents a regular folder in the server database.
ManagedFolderInformationType	Contains information about a managed custom folder.
PermissionSetType	Contains all the permissions that are configured for a folder.
PermissionType	Specifies a permission on a folder.
TargetFolderIdType	Specifies a target folder for operations that create, save, copy, move, or synchronize items or folders.

2.2.4.1 t:AddressListIdType

The **AddressListIdType** complex type specifies an identifier of an address list. This type extends the **BaseFolderIdType** complex type, as specified in section 2.2.4.4.<1>

```
<xs:complexType name="AddressListIdType">
  <xs:complexContent>
    <xs:extension base="t:BaseFolderIdType">
      <xs:attribute name="Id" type="xs:string" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the attribute of the **AddressListIdType** complex type.

Attribute name	Type	Description
Id	**xs:string** ([XMLSCHEMA2] section 3.2.1)	The identifier of the address list.

2.2.4.2 t:ArrayOfFoldersType

The **ArrayOfFoldersType** complex type specifies an array of folders that are used in folder operations.

```
<xs:complexType name="ArrayOfFoldersType">
  <xs:choice
    minOccurs="0"
    maxOccurs="unbounded"
  >
    <xs:element name="Folder"
      type="t:FolderType"
     />
    <xs:element name="CalendarFolder"
      type="t:CalendarFolderType"
     />
    <xs:element name="ContactsFolder"
      type="t:ContactsFolderType"
     />
    <xs:element name="SearchFolder"
      type="t:SearchFolderType"
     />
    <xs:element name="TasksFolder"
      type="t:TasksFolderType"
     />
  </xs:choice>
</xs:complexType>
```

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

Element name	Type	Description
Folder	**t:FolderType** (section 2.2.4.12)	Represents a regular folder in the server database.
CalendarFolder	**t:CalendarFolderType** ([MS-OXWSMTGS] section 2.2.4.9)	Represents a **Calendar folder** in a mailbox.
ContactsFolder	**t:ContactsFolderType** ([MS-OXWSCONT] section 3.1.4.1.1.6)	Represents a **Contacts folder** in a mailbox.
SearchFolder	**t:SearchFolderType** ([MS-OXWSSRCH] section 2.2.3.26)	Represents a **search folder** that is contained in a mailbox.
TasksFolder	**t:TasksFolderType** ([MS-OXWSTASK] section 2.2.4.5)	Represents a **Tasks folder** that is contained in a mailbox.

2.2.4.3 t:ArrayOfUnknownEntriesType

The **ArrayOfUnknownEntriesType** complex type represents an array of unknown permission entries that cannot be resolved against the directory service. The text value represents a security identifier (SID).

```
<xs:complexType name="ArrayOfUnknownEntriesType">
  <xs:choice
    minOccurs="0"
    maxOccurs="unbounded"
  >
    <xs:element name="UnknownEntry"
      type="xs:string"
     />
  </xs:choice>
</xs:complexType>
```

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

Element name	Type	Description
UnknownEntry	**xs:string** [XMLSCHEMA2]	Represents an array of unknown permission entries that cannot be resolved against the directory service.

2.2.4.4 t:BaseFolderIdType

The **BaseFolderIdType** complex type specifies the base type for derived types that represent a folder identifier.

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

The **DistinguishedFolderIdType** complex type ([MS-OXWSCDATA] section 2.2.4.27) and the **FolderIdType** complex type ([MS-OXWSCDATA] section 2.2.4.36) extend the **BaseFolderIdType** complex type.

2.2.4.5 t:BaseFolderType

The **BaseFolderType** complex type specifies the base type for folders.

```
<xs:complexType name="BaseFolderType"
  abstract="true"
>
  <xs:sequence>
    <xs:element name="FolderId"
      type="t:FolderIdType"
      minOccurs="0"
     />
    <xs:element name="ParentFolderId"
      type="t:FolderIdType"
      minOccurs="0"
     />
    <xs:element name="FolderClass"
      type="xs:string"
```

```
        minOccurs="0"
       />
      <xs:element name="DisplayName"
        type="xs:string"
        minOccurs="0"
       />
      <xs:element name="TotalCount"
        type="xs:int"
        minOccurs="0"
       />
      <xs:element name="ChildFolderCount"
        type="xs:int"
        minOccurs="0"
       />
      <xs:element name="ExtendedProperty"
        type="t:ExtendedPropertyType"
        minOccurs="0"
        maxOccurs="unbounded"
       />
      <xs:element name="ManagedFolderInformation"
        type="t:ManagedFolderInformationType"
        minOccurs="0"
       />
      <xs:element name="EffectiveRights"
        type="t:EffectiveRightsType"
        minOccurs="0"
       />
      <xs:element name="DistinguishedFolderId"
        type="t:DistinguishedFolderIdNameType"
        minOccurs="0"
       />
      <xs:element name="PolicyTag"
        type="t:RetentionTagType"
        minOccurs="0"
       />
      <xs:element name="ArchiveTag"
        type="t:RetentionTagType"
        minOccurs="0"
       />
      <xs:element name="ReplicaList"
        type="t:ArrayOfStringsType"
        minOccurs="0"
       />
    </xs:sequence>
  </xs:complexType>
```

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

Element name	Type	Description
FolderId	**t:FolderIdType** ([MS-OXWSCDATA] section 2.2.4.36)	Specifies the folder identifier and change key. This element can be present and cannot be set during folder creation. The maximum length for the **FolderIdType** element **Id** attribute and the maximum length for the **FolerIdType ChangeKey** attribute is 512 bytes after **base64** decoding.
ParentFolderId	**t:FolderIdType**	Specifies the folder identifier and change key for the parent folder. This element can be present and cannot be set during folder creation. The maximum length for the

Element name	Type	Description
		FolderIdType element **Id** attribute and the maximum length for the **FolerIdType ChangeKey** attribute is 512 bytes after base64 decoding.
FolderClass	**xs:string** ([XMLSCHEMA2])	Specifies the folder class. This value MUST be "IPF.Appointment" for Calendar folders, "IPF.Contact" for Contacts folders, "IPF.Journal" for journal folders, "IPF.Note" for mail folders, "IPF.StickyNote" for note folders, and "IPF.Task" for Tasks folders. The folders class can be a custom class. The following example shows how to set a custom folder class: "IPF.<folderclass>", where <folderclass> is the name of the custom class. This element can be present.
DisplayName	**xs:string**	Specifies the display name of the folder. This element can be present. This element is required in a **CreateFolder** operation request.
TotalCount	**xs:int** ([XMLSCHEMA2])	Specifies the total number of items in a folder. This property MUST be read-only for a client and is returned only in a response. This element can be present.
ChildFolderCount	**xs:int**	Specifies the total number of child folders in a folder. This property MUST be read-only for a client and is returned only in a response. This element can be present.
ExtendedProperty	**t:ExtendedPropertyType** ([MS-OXWSXPROP] section 2.1.5)	Specifies the set of extended properties on a folder. This element can be present.
ManagedFolderInformation	**t:ManagedFolderInformationType** (section 2.2.4.13)	Specifies metadata for a managed folder. This property MUST be read-only for a client and is only returned in a response. This element can be present.
EffectiveRights	**t:EffectiveRightsType** ([MS-OXWSCDATA] section 2.2.4.29)	Specifies the operations that the user that is logged user can perform on the folder. This property MUST be read-only for a client and is returned only in a response. This element can be present.
DistinguishedFolderId	**t:DistinguishedFolderIdNameType** ([MS-OXWSCDATA] section 2.2.5.10)	Specifies an identifier for a folder that can be referenced by name. This element only supports "conversationhistory", "adminauditlogs", and "recoverableitemsroot" when it is present in a **Folder**, **CalendarFolder**, **ContactsFolder**, or **TasksFolder** element of

Element name	Type	Description
		NonEmptyArrayOfFoldersType complex type in **CreateFolder** operation.<2>
PolicyTag	**t:RetentionTagType** ([MS-OXWSURPT] section 2.2.4.1)	Specifies the retention policy.<3>
ArchiveTag	**t:RetentionTagType**	Specifies that the item is archived.<4>
ReplicaList	**t:ArrayofStringsType** ([MS-OXWSCDATA] section 2.2.4.13)	A list of replicas of a public folder.<5>

The **FolderType** complex type, as specified in section 2.2.4.12, **CalendarFolderType** complex type, as specified in [MS-OXWSMTGS] section 2.2.4.9, and **ContactsFolderType** complex type, as specified in [MS-OXWSCONT] section 3.1.4.1.1.6, extend the **BaseFolderType** complex type.

2.2.4.6 m:BaseMoveCopyFolderType

The **BaseMoveCopyFolderType** complex type is the base type for the **CopyFolderType** complex type, as specified in section 3.1.4.1.3.2, and the **MoveFolderType** complex type, as specified in section 3.1.4.7.3.2. The **BaseMoveCopyFolderType** complex type extends the **BaseRequestType** complex type ([MS-OXWSCDATA] section 2.2.4.17).

```
<xs:complexType name="BaseMoveCopyFolderType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="ToFolderId"
          type="t:TargetFolderIdType"
         />
        <xs:element name="FolderIds"
          type="t:NonEmptyArrayOfBaseFolderIdsType"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
ToFolderId	**t:TargetFolderIdType** (section 2.2.4.16)	Represents a target folder for operations that create, save, copy, move, or synchronize items or folders.
FolderIds	**t:NonEmptyArrayOfBaseFolderIdsType** (section 3.1.4.6.3.3)	Specifies an array of one or more folder identifiers.

2.2.4.7 t:BasePermissionType

The **BasePermissionType** complex type is an abstract type that defines base permissions for items and folders.

```
<xs:complexType name="BasePermissionType"
  abstract="true"
>
  <xs:sequence>
    <xs:element name="UserId"
      type="t:UserIdType"
      maxOccurs="1"
      minOccurs="1"
     />
    <xs:element name="CanCreateItems"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="CanCreateSubFolders"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="IsFolderOwner"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="IsFolderVisible"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="IsFolderContact"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="EditItems"
      type="t:PermissionActionType"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="DeleteItems"
      type="t:PermissionActionType"
      minOccurs="0"
      maxOccurs="1"
     />
  </xs:sequence>
</xs:complexType>
```

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

Element name	Type	Description
UserId	**t:UserIdType** ([MS-OXWSCDATA] section 2.2.4.62)	Specifies a user identifier. This element MUST be present.
CanCreateItems	**xs:boolean** [XMLSCHEMA2]	Specifies whether a client can create items in a folder. A value of "true" indicates that the client can create an item. **IsFolderOwner** MUST also be "true" for the client. This element can be

Element name	Type	Description
		present.
CanCreateSubFolders	**xs:boolean**	Specifies whether the client can create subfolders. A value of "true" indicates that the client can create a subfolder. **IsFolderOwner** MUST also be "true" for the client. This element can be present.
IsFolderOwner	**xs:boolean**	Specifies whether the user is the owner of a folder. A value of "true" indicates that the user is the owner of the folder. This element can be present.
IsFolderVisible	**xs:boolean**	Specifies whether a user can view a folder. A value of "true" indicates that the folder is visible. This element can be present.
IsFolderContact	**xs:boolean**	Specifies whether a user is a **contact** for a folder. A value of "true" indicates that the user is the contact for the folder. This element can be present.
EditItems	**t:PermissionActionType** (section 2.2.5.2)	Specifies whether a client can edit items in the folder. **IsFolderOwner** MUST also be "true" for the client. This element can be present.
DeleteItems	**t:PermissionActionType**	Specifies whether the client can delete items in the folder. **IsFolderOwner** MUST also be "true" for the client. This element can be present.

The **PermissionType** complex type, as specified in section 2.2.4.15, and the **CalendarPermissionType** complex type, as specified in [MS-OXWSMTGS] section 2.2.4.12, extend the **BasePermissionType** complex type.

2.2.4.8 t:ConsumerCalendarIdType

The **ConsumerCalendarIdType** complex type specifies the identifier of a consumer calendar folder.<6> This complex type extends the **BaseFolderIdType** complex type, as specified in section 2.2.4.5.

```
<xs:complexType name="ConsumerCalendarIdType">
  <xs:complexContent>
    <xs:extension base="t:BaseFolderIdType">
      <xs:attribute name="OwnerPuid" type="xs:long" use="required"/>
      <xs:attribute name="OwnerCid" type="xs:long" use="optional"/>
      <xs:attribute name="CalendarGuid" type="t:GuidType" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element Name	Type	Description
OwnerPuid	**xs:Long** ([XMLSCHEMA2])	Specifies the **PUID** of the folder owner.
OwnerCid	**xs:Long**	Specifies the **contact identifier**

Element Name	Type	Description
		(CID) of the folder owner.
CalendarGuid	**t:GuidType** ([MS-OXWSXPROP] section 2.1.7)	Specifies the **GUID** of the calendar.

2.2.4.9 t:FolderChangeDescriptionType

The **FolderChangeDescriptionType** complex type specifies a change to a single folder property. The **FolderChangeDescriptionType** complex type extends the **ChangeDescriptionType** complex type, as specified in [MS-OXWSCORE] section 3.1.4.9.3.4.

```
<xs:complexType name="FolderChangeDescriptionType">
  <xs:complexContent>
    <xs:extension
      base="t:ChangeDescriptionType"
     />
  </xs:complexContent>
</xs:complexType>
```

The **SetFolderFieldType** complex type, as specified in section 3.1.4.8.3.7, the **DeleteFolderFieldType** complex type, as specified in section 3.1.4.8.3.4, and the **AppendToFolderFieldType** complex type, as specified in section 3.1.4.8.3.3 extend the **FolderChangeDescriptionType** complex type.

2.2.4.10 t:FolderChangeType

The **FolderChangeType** complex type specifies a collection of changes to be performed on a single folder.

```
<xs:complexType name="FolderChangeType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="FolderId"
        type="t:FolderIdType"
       />
      <xs:element name="DistinguishedFolderId"
        type="t:DistinguishedFolderIdType"
       />
    </xs:choice>
    <xs:element name="Updates"
      type="t:NonEmptyArrayOfFolderChangeDescriptionsType"
     />
  </xs:sequence>
</xs:complexType>
```

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

Element name	Type	Description
FolderId	**t:FolderIdType** ([MS-OXWSCDATA] section 2.2.4.36)	Specifies the folder identifier and change key. The maximum length for the **FolderIdType**

Element name	Type	Description
		element **Id** attribute and the maximum length for the **FolerIdType ChangeKey** attribute is 512 bytes after base64 decoding.
DistinguishedFolderId	**t:DistinguishedFolderIdType** ([MS-OXWSCDATA] section 2.2.4.27)	Specifies an identifier for a distinguished folder.
Updates	**t:NonEmptyArrayOfFolderChangeDescriptionsType** (section 3.1.4.8.3.5)	Specifies a collection of changes to a folder.

2.2.4.11 m:FolderInfoResponseMessageType

The **FolderInfoResponseMessageType** complex type represents the response message for the **CreateFolder** operation, as specified in section 3.1.4.2, the **GetFolder** operation, as specified in section 3.1.4.6, the **UpdateFolder** operation, as specified in section 3.1.4.8, the **MoveFolder** operation, as specified in section 3.1.4.7, the **CopyFolder** operation, as specified in section 3.1.4.1, and the **CreateManagedFolder** operation, as specified in section 3.1.4.3. The **FolderInfoResponseMessageType** complex type extends the **ResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.4.67).

```
<xs:complexType name="FolderInfoResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="Folders"
          type="t:ArrayOfFoldersType"
          minOccurs="0"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
Folders	**t:ArrayOfFoldersType** (section 2.2.4.2)	Represents the folders that are returned with the response message.

2.2.4.12 t:FolderType

The **FolderType** complex type represents a regular folder in the server database. The **FolderType** complex type extends the **BaseFolderType** complex type, as specified in section 2.2.4.5.

```
<xs:complexType name="FolderType">
```

```
  <xs:complexContent>
    <xs:extension
      base="t:BaseFolderType"
    >
      <xs:sequence>
        <xs:element name="PermissionSet"
          type="t:PermissionSetType"
          minOccurs="0"
         />
        <xs:element name="UnreadCount"
          type="xs:int"
          minOccurs="0"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
PermissionSet	**t:PermissionSetType** (section 2.2.4.14)	Specifies all permissions that are configured for a folder. This element can be present.
UnreadCount	**xs:int** ([XMLSCHEMA2])	Specifies the number of unread items in a folder. This element MUST only exist in responses and MUST equal the sum of all **MessageType** complex types ([MS-OXWSMSG] section 2.2.4.1) and **PostItemType** complex types ([MS-OXWSPOST] section 2.2.4.1) that have the **IsRead** property set to "false". This element can be present.

The **SearchFolderType** complex type ([MS-OXWSSRCH] section 2.2.3.26) and the **TasksFolderType** complex type ([MS-OXWSTASK] section 2.2.4.5) extend the **FolderType** complex type.

2.2.4.13 t:ManagedFolderInformationType

The **ManagedFolderInformationType** complex type contains information about a managed custom folder.

```
<xs:complexType name="ManagedFolderInformationType">
  <xs:sequence>
    <xs:element name="CanDelete"
      type="xs:boolean"
      minOccurs="0"
     />
    <xs:element name="CanRenameOrMove"
      type="xs:boolean"
      minOccurs="0"
     />
    <xs:element name="MustDisplayComment"
      type="xs:boolean"
      minOccurs="0"
     />
    <xs:element name="HasQuota"
      type="xs:boolean"
      minOccurs="0"
     />
    <xs:element name="IsManagedFoldersRoot"
      type="xs:boolean"
      minOccurs="0"
```

```
      />
    <xs:element name="ManagedFolderId"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="Comment"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="StorageQuota"
      type="xs:int"
      minOccurs="0"
     />
    <xs:element name="FolderSize"
      type="xs:int"
      minOccurs="0"
     />
    <xs:element name="HomePage"
      type="xs:string"
      minOccurs="0"
     />
  </xs:sequence>
</xs:complexType>
```

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

Element name	Type	Description
CanDelete	**xs:boolean** ([XMLSCHEMA2])	A value that indicates whether a managed folder can be deleted by a client. A value of "true" indicates that the managed folder can be deleted. This element can be present.
CanRenameOrMove	**xs:boolean**	A value that indicates whether a managed folder can be renamed or moved by a client. A value of "true" indicates that the managed folder can be renamed or moved. This element can be present.
MustDisplayComment	**xs:boolean**	A value that indicates whether the managed folder comment is to be displayed. A value of "true" indicates that the client MUST display the managed folder comment. This element can be present.
HasQuota	**xs:boolean**	A value that indicates whether a managed folder has a quota. A value of "true" indicates that the **StorageQuota** property was serialized into the SOAP response. This element can be present.
IsManagedFoldersRoot	**xs:boolean**	A value that indicates whether the managed folder is the root managed folder. A value of "true" indicates that the managed is the root managed folder. This element can be present.
ManagedFolderId	**xs:string** ([XMLSCHEMA2])	The unique identifier of a managed folder. This element can be present.
Comment	**xs:string**	A comment that is associated with a managed folder. This element can be present.
StorageQuota	**xs:int** ([XMLSCHEMA2])	The storage quota for a managed folder. This element can be present.
FolderSize	**xs:int**	A value that describes the total size of all the contents of a managed folder. This element can be present.

Element name	Type	Description
HomePage	**xs:string**	The URL that is the default home page for the managed folder. This element can be present.

2.2.4.14 t:PermissionSetType

The **PermissionSetType** complex type contains all the permissions that are configured for a folder. This constitutes the set of permissions on a folder.

```
<xs:complexType name="PermissionSetType">
  <xs:sequence>
    <xs:element name="Permissions"
      type="t:ArrayOfPermissionsType"
     />
    <xs:element name="UnknownEntries"
      type="t:ArrayOfUnknownEntriesType"
      minOccurs="0"
     />
  </xs:sequence>
</xs:complexType>
```

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

Element name	Type	Description
Permissions	**t:ArrayOfPermissionsType** ([MS-OXWSCDATA] section 2.2.4.9)	Specifies a collection of permissions for a folder.
UnknownEntries	**t:ArrayOfUnknownEntriesType** (section 2.2.4.3)	Represents an array of unknown permission entries that cannot be resolved against the directory service. The text value represents a security identifier (SID). This element can be present.

2.2.4.15 t:PermissionType

The **PermissionType** complex type specifies a permission on a folder. The **PermissionType** complex type extends the **BasePermissionType** complex type, as specified in section 2.2.4.7.<7>

```
<xs:complexType name="PermissionType">
  <xs:complexContent>
    <xs:extension
      base="t:BasePermissionType"
    >
      <xs:sequence>
        <xs:element name="ReadItems"
          type="t:PermissionReadAccessType"
          minOccurs="0"
          maxOccurs="1"
         />
        <xs:element name="PermissionLevel"
          type="t:PermissionLevelType"
          minOccurs="1"
          maxOccurs="1"
```

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

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

Element name	Type	Description
ReadItems	**t:PermissionReadAccessType** (section 2.2.5.4)	Specifies whether a user has permission to read items in a folder. This element can be present.
PermissionLevel	**t:PermissionLevelType** (section 2.2.5.3)	Specifies the combination of permissions that a user has on a folder. This element MUST be present.

This type defines the access that a user has to a folder.

2.2.4.16 t:TargetFolderIdType

The **TargetFolderIdType** complex type specifies a target folder for operations that create, save, copy, move, or synchronize items or folders.

```
<xs:complexType name="TargetFolderIdType">
  <xs:choice>
    <xs:element name="FolderId" type="t:FolderIdType"/>
    <xs:element name="DistinguishedFolderId" type="t:DistinguishedFolderIdType"/>
    <xs:element name="AddressListId" type="t:AddressListIdType"/>
    <xs:element name="ConsumerCalendarId" type="t:ConsumerCalendarIdType"/>
  </xs:choice>
</xs:complexType>
```

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

Element name	Type	Description
FolderId	**t:FolderIdType** ([MS-OXWSCDATA] section 2.2.4.36)	Specifies a folder identifier. The maximum length for the **FolderIdType** element **Id** attribute and the maximum length for the **FolerIdType ChangeKey** attribute is 512 bytes after base64 decoding.
DistinguishedFolderId	**t:DistinguishedFolderIdType** ([MS-OXWSCDATA] section 2.2.4.27)	Specifies a distinguished folder identifier.
AddressListId	**t:AddressListIdType** (section 2.2.4.1)	Specifies the identifier of an address list.
ConsumerCalendarId	**t:ConsumerCalendarIdType** (section 2.2.4.8)	Specifies the identifier of a consumer calendar.<8>

This type is a container for either a **FolderIdType** complex type ([MS-OXWSCDATA] section 2.2.4.36) or **DistinguishedFolderIdType** complex type ([MS-OXWSCDATA] section 2.2.4.27).

2.2.5 Simple Types

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

Simple type names	Description
FolderClassType	Represents the folder.
PermissionActionType	Defines which items in a folder a user has permission to edit or delete.
PermissionLevelType	Specifies the permission level that a user has on a folder.
PermissionReadAccessType	Specifies whether a user has permission to read items in a folder.

2.2.5.1 t:FolderClassType

The **FolderClassType** simple type represents the folder.

```
<xs:simpleType name="FolderClassType">
  <xs:restriction
    base="xs:string"
   />
</xs:simpleType>
```

2.2.5.2 t:PermissionActionType

The **PermissionActionType** simple type defines which items in a folder a user has permission to edit or delete.

```
<xs:simpleType name="PermissionActionType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="None"
     />
    <xs:enumeration
      value="Owned"
     />
    <xs:enumeration
      value="All"


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

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

Value	Meaning
None	Indicates that the user does not have permission to perform the action on any items in the folder.

Value	Meaning
Owned	Indicates that the user has permission to perform the action on the items in the folder that the user owns.
All	Indicates that the user has permission to perform the action on all items in the folder.

2.2.5.3 t:PermissionLevelType

The **PermissionLevelType** simple type specifies the permission level that a user has on a folder.

```
<xs:simpleType name="PermissionLevelType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="None"
     />
    <xs:enumeration
      value="Owner"
     />
    <xs:enumeration
      value="PublishingEditor"
     />
    <xs:enumeration
      value="Editor"
     />
    <xs:enumeration
      value="PublishingAuthor"
     />
    <xs:enumeration
      value="Author"
     />
    <xs:enumeration
      value="NoneditingAuthor"
     />
    <xs:enumeration
      value="Reviewer"
     />
    <xs:enumeration
      value="Contributor"
     />
    <xs:enumeration
      value="Custom"
     />
  </xs:restriction>
</xs:simpleType>
```

The following table list the values that are defined by the **PermissionLevelType** simple type.

Value	Meaning
None	The user has no permissions on the folder.
Owner	The user can create, read, edit, and delete all items in the folder and create subfolders. The user is both folder owner and folder contact.
PublishingEditor	The user can create, read, edit, and delete all items in the folder, and create subfolders.
Editor	The user can create, read, edit, and delete all items in the folder.

Value	Meaning
PublishingAuthor	The user can create and read all items in the folder, edit and delete only items that the user creates, and create subfolders.
Author	The user can create and read all items in the folder, and edit and delete only items that the user creates.
NoneditingAuthor	The user can create and read all items in the folder, and delete only items that the user creates.
Reviewer	The user can read all items in the folder.
Contributor	The user can create items in the folder but cannot read any items in the folder.<9>
Custom	The user has custom access permissions on the folder.

2.2.5.4 t:PermissionReadAccessType

The **PermissionReadAccessType** simple type specifies whether a user has permission to read items in a folder.

```
<xs:simpleType name="PermissionReadAccessType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="None"
     />
    <xs:enumeration
      value="FullDetails"
     />
  </xs:restriction>
</xs:simpleType>
```

The following table list the values that are defined by the **PermissionReadAccessType** simple type.

Value	Description
None	The user does not have permission to read items in the folder.
FullDetails	The user has permission to read all items in the folder.

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3 Protocol Details

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

3.1 ExchangeServicePortType Server Details

The Folders and Folder Permissions Web Service Protocol defines a single port type with eight operations. The operations enable client implementations to copy, create, delete, empty, get, move, and update folders and managed folders.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

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

Operation name	Description
CopyFolder	Copies an identified folder and returns the folder identifier and change key of the copied folder.
CreateFolder	Creates folders, Calendar folders, Contacts folders, Tasks folders, and search folders.
CreateManagedFolder	Creates a managed folder in the server **message store**.
DeleteFolder	Deletes folders from a mailbox.
EmptyFolder	Empties folders and enables deletion of the subfolders from a mailbox.
GetFolder	Gets folders from the server message store.
MoveFolder	Moves folders from a specified folder and puts them in another folder.
UpdateFolder	Modifies properties of an existing item in the server message store.

3.1.4.1 CopyFolder

The **CopyFolder** operation copies an identified folder and returns the folder identifier and change key of the copied folder.<10>

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

```
<wsdl:operation name="CopyFolder">
    <wsdl:input message="tns:CopyFolderSoapIn" />
    <wsdl:output message="tns:CopyFolderSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification of the **CopyFolder** operation.

```
<wsdl:operation name="CopyFolder">
    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CopyFolder" />
    <wsdl:input>
        <soap:header message="tns:CopyFolderSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:CopyFolderSoapIn" part="MailboxCulture" use="literal"/>
        <soap:header message="tns:CopyFolderSoapIn" part="RequestVersion" use="literal"/>
        <soap:body parts="request" use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="CopyFolderResult" use="literal" />
        <soap:header message="tns:CopyFolderSoapOut" part="ServerVersion" use="literal"/>
    </wsdl:output>
</wsdl:operation>
```

The protocol client sends a **CopyFolderSoapIn** request WSDL message, and the protocol server responds with a **CopyFolderSoapOut** response WSDL message.

A successful **CopyFolder** operation request returns a **CopyFolderResponse** element with the **ResponseClass** attribute of the **CopyFolderResponseMessage** element set to "Success" and the **ResponseCode** element of the **CopyFolderResponse** element set to "NoError".

An unsuccessful **CopyFolder** operation request returns a **CopyFolderResponse** element with the **ResponseClass** attribute of the **CopyFolderResponseMessage** element set to "Error". The **ResponseCode** element of the **CopyFolderResponseMessage** element is set to a value of the **ResponseCodeType** simple type, as specified in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.1.1 Messages

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

Message name	Description
CopyFolderSoapIn	Specifies the **SOAP message** that copies a folder.
CopyFolderSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.1.1.1 tns:CopyFolderSoapIn Message

The **CopyFolderSoapIn** WSDL message specifies the **CopyFolder** operation request to copy a folder.

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

```
</wsdl:message>
```

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

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

Part name	Element/type	Description
request	**tns:CopyFolder** (section 3.1.4.1.2.1)	Specifies the **SOAP body** of the request to copy a folder.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.4.33)	Specifies a **SOAP header** that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.4.45)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the **CopyFolder** operation request.

3.1.4.1.1.2 tns:CopyFolderSoapOut Message

The **CopyFolderSoapOut** WSDL message specifies the server response to the **CopyFolder** operation request to copy a folder.

```
<wsdl:message name="CopyFolderSoapOut">
    <wsdl:part name="CopyFolderResult" element="tns:CopyFolderResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

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

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

Part name	Element/type	Description
CopyFolderResult	**tns:CopyFolderResponse** (section 3.1.4.1.2.2)	Specifies the SOAP body of the response to a **CopyFolder** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.1.2 Elements

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

Element name	Description
CopyFolder	Specifies a request to copy a folder in a mailbox in the server message store.

Element name	Description
CopyFolderResponse	Specifies the response body content from a request to copy a folder.

3.1.4.1.2.1 CopyFolder Element

The **CopyFolder** element specifies a request message for a **CopyFolder** operation.

```
<xs:element name="CopyFolder"
  type="m:CopyFolderType"
 />
```

3.1.4.1.2.2 CopyFolderResponse Element

The **CopyFolderResponse** element specifies a response message for a **CopyFolder** operation.

```
<xs:element name="CopyFolderResponse"
  type="m:CopyFolderResponseType"
 />
```

3.1.4.1.3 Complex Types

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

Complex type name	Description
CopyFolderResponseType	Specifies a response message for the **CopyFolder** operation.
CopyFolderType	Specifies a request message for the **CopyFolder** operation.

3.1.4.1.3.1 m:CopyFolderResponseType Complex Type

The **CopyFolderResponseType** complex type specifies the response message that is returned by the **CopyFolder** operation. The **CopyFolderResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.18.

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

3.1.4.1.3.2 m:CopyFolderType Complex Type

The **CopyFolderType** complex type specifies a request message to copy folders in a server database. The **CopyFolderType** complex type extends the **BaseMoveCopyFolderType** complex type, as specified in section 2.2.4.6.

```
<xs:complexType name="CopyFolderType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseMoveCopyFolderType"
     />
  </xs:complexContent>
</xs:complexType>
```

3.1.4.2 CreateFolder

The **CreateFolder** operation creates folders, Calendar folders, Contacts folders, Tasks folders, and search folders.

It is recommended that before any data is read from or written to a folder, an implementation ensures that the folder exists and opens it, or creates it if it does not exist. Before a folder can be created, the parent folder MUST already exist. Trying to create a folder that already exists results in an error.

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

```
<wsdl:operation name="CreateFolder">
    <wsdl:input message="tns:CreateFolderSoapIn" />
    <wsdl:output message="tns:CreateFolderSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification of the **CreateFolder** operation.

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

The protocol client sends a **CreateFolderSoapIn** request WSDL message, and the protocol server responds with a **CreateFolderSoapOut** response WSDL message.

A successful **CreateFolder** operation request returns a **CreateFolderResponse** element with the **ResponseClass** attribute of the **CreateFolderResponseMessage** element set to "Success" and the **ResponseCode** element of the **CreateFolderResponse** element set to "NoError".

An unsuccessful **CreateFolder** operation request returns a **CreateFolderResponse** element with the **ResponseClass** attribute of the **CreateFolderResponseMessage** element set to "Error". The **ResponseCode** element of the **CreateFolderResponseMessage** element is set to a value of the **ResponseCodeType** simple type, as specified in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.2.1 Messages

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

Messages name	Description
CreateFolderSoapIn	Specifies the SOAP message that creates a folder.
CreateFolderSoapOut	Specifies the SOAP message that is returned by the server in SOAP message that is returned by the server in response.

3.1.4.2.1.1 tns:CreateFolderSoapIn Message

The **CreateFolderSoapIn** WSDL message specifies the **CreateFolder** operation request to create a new folder.

```
<wsdl:message name="CreateFolderSoapIn">
    <wsdl:part name="request" element="tns:CreateFolder" />
    <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>
```

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

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

Part name	Element/type	Description
request	**tns:CreateFolder** (section 3.1.4.2.2.1)	Specifies the SOAP body of the request to create a folder.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.3.7)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the **CreateFolder** operation request.
TimeZoneContext	**t:TimeZoneContext** ([MS-OXWSGTZ] section 2.2.3.4)	Specifies a time zone definition and enables the association of SOAP attributes with the definition.

3.1.4.2.1.2 tns:CreateFolderSoapOut Message

The **CreateFolderSoapOut** WSDL message specifies the server response to the **CreateFolder** operation request to create a folder.

```
<wsdl:message name="CreateFolderSoapOut">
    <wsdl:part name="CreateFolderResult" element="tns:CreateFolderResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

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

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

Part name	Element/type	Description
CreateFolderResult	**tns:CreateFolderResponse** (section 3.1.4.2.2.2)	Specifies the SOAP body of the response to a **CreateFolder** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.2.2 Elements

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

Element name	Description
CreateFolder	Specifies a request to create folders in a mailbox in the server message store.
CreateFolderResponse	Specifies the response body content from a request to create a folder.

3.1.4.2.2.1 CreateFolder Element

The **CreateFolder** element specifies a request message for a **CreateFolder** operation.

```
<xs:element name="CreateFolder"
  type="m:CreateFolderType"
 />
```

3.1.4.2.2.2 CreateFolderResponse Element

The **CreateFolderResponse** element specifies a response message for a **CreateFolder** operation.

```
<xs:element name="CreateFolderResponse"
  type="m:CreateFolderResponseType"
 />
```

3.1.4.2.3 Complex Types

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

Complex type name	Description
CreateFolderResponseType	Specifies a response message for the **CreateFolder** operation.
CreateFolderType	Specifies a request message for the **CreateFolder** operation.
NonEmptyArrayOfFoldersType	Specifies an array of folders that are used in folder operations. This array has at least one member.

3.1.4.2.3.1 m:CreateFolderResponseType Complex Type

The **CreateFolderResponseType** complex type specifies the response message that is returned by **CreateFolder** operation. The **CreateFolderResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.18.

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

3.1.4.2.3.2 m:CreateFolderType Complex Type

The **CreateFolderType** complex type specifies a request message to create a folder in the server database. The **CreateFolderType** complex type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.17.

```
<xs:complexType name="CreateFolderType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="ParentFolderId"
          type="t:TargetFolderIdType"
         />
        <xs:element name="Folders"
          type="t:NonEmptyArrayOfFoldersType"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
ParentFolderId	**t:TargetFolderIdType** (section 2.2.4.16)	The identifier of the folder that will contain the newly created folder.
Folders	**t:NonEmptyArrayOfFoldersType** (section 3.1.4.2.3.3)	Represents an array of folders to be created. This array has at least one member.

3.1.4.2.3.3 t:NonEmptyArrayOfFoldersType Complex Type

The **NonEmptyArrayOfFoldersType** complex type represents an array of folders that has at least one member.

```
<xs:complexType name="NonEmptyArrayOfFoldersType">
  <xs:choice
    minOccurs="1"
    maxOccurs="unbounded"
  >
    <xs:element name="Folder"
      type="t:FolderType"
     />
    <xs:element name="CalendarFolder"
      type="t:CalendarFolderType"
     />
    <xs:element name="ContactsFolder"
      type="t:ContactsFolderType"
     />
    <xs:element name="SearchFolder"
      type="t:SearchFolderType"
     />
    <xs:element name="TasksFolder"
      type="t:TasksFolderType"
     />
  </xs:choice>
</xs:complexType>
```

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

Element	Type	Description
Folder	**t:FolderType** (section 2.2.4.12)	Represents a regular folder in the server database.
CalendarFolder	**t:CalendarFolderType** ([MS-OXWSMTGS] section 2.2.4.9)	Represents a folder that primarily contains calendar items.
ContactsFolder	**t:ContactsFolderType** ([MS-OXWSCONT] section 3.1.4.1.1.6)	Represents a Contacts folder in a mailbox.
SearchFolder	**t:SearchFolderType** ([MS-OXWSSRCH] section 2.2.4.26)	Represents a search folder that is contained in a mailbox.
TasksFolder	**t:TasksFolderType** ([MS-OXWSTASK] section 2.2.4.5)	Represents a Tasks folder that is contained in a mailbox.

3.1.4.3 CreateManagedFolder

The **CreateManagedFolder** operation creates a **managed folder** in the server message store.

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

```
<wsdl:operation name="CreateManagedFolder">
    <wsdl:input message="tns:CreateManagedFolderSoapIn" />
    <wsdl:output message="tns:CreateManagedFolderSoapOut" />
```

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

The following is the WSDL binding specification of the **CreateManagedFolder** operation.

```
<wsdl:operation name="CreateManagedFolder">
    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CreateManagedFolder"
/>
    <wsdl:input>
        <soap:header message="tns:CreateManagedFolderSoapIn" part="Impersonation"
use="literal"/>
        <soap:header message="tns:CreateManagedFolderSoapIn" part="MailboxCulture"
use="literal"/>
        <soap:header message="tns:CreateManagedFolderSoapIn" part="RequestVersion"
use="literal"/>
        <soap:body parts="request" use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="CreateManagedFolderResult" use="literal" />
        <soap:header message="tns:CreateManagedFolderSoapOut" part="ServerVersion"
use="literal"/>
    </wsdl:output>
</wsdl:operation>
```

A managed folder is created by an administrator and placed in a user's mailbox for messaging records management purposes. The retention and journaling of messages in managed folders are controlled by managed content settings that are applied to the folder.

The protocol client sends a **CreateManagedFolderSoapIn** request WSDL message, and the protocol server responds with a **CreateManagedSoapOut** response WSDL message.

A successful **CreateManagedFolder** operation request returns a **CreateManagedFolderResponse** element with the **ResponseClass** attribute of the **CreateManagedFolderResponseMessage** element set to "Success" and the **ResponseCode** element of the **CreateManagedFolderResponse** element set to "NoError".

An unsuccessful **CreateManagedFolder** operation request returns a **CreateManagedFolderResponse** element with the **ResponseClass** attribute of the **CreateManagedFolderResponseMessage** element set to "Error". The **ResponseCode** element of the **CreateManagedFolderResponseMessage** element is set to a value of the **ResponseCodeType** simple type, as specified in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.3.1 Messages

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

Message name	Description
CreateManagedFolderSoapIn	Specifies the request that creates a managed folder.
CreateManagedFolderSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.3.1.1 tns:CreateManagedFolderSoapIn Message

The **CreateManagedFolderSoapIn** WSDL message specifies the **CreateManagedFolder** operation request to create a managed folder.

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

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

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

Part name	Element/type	Description
request	**tns:CreateManagedFolder** (section 3.1.4.3.2.1)	Specifies the SOAP body of the request to create a managed folder.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.3.7)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the **CreateManagedFolder** operation request.

3.1.4.3.1.2 tns:CreateManagedFolderSoapOut Message

The **CreateManagedFolderSoapOut** WSDL message specifies the server response to the **CreateManagedFolder** operation request to create a managed folder.

```
<wsdl:message name="CreateManagedFolderSoapOut">
  <wsdl:part name="CreateManagedFolderResult" element="tns:CreateManagedFolderResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

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

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

Part name	Element/type	Description
CreateManagedFolderResult	**tns:CreateManagedFolderResponse** (section 3.1.4.3.2.2)	Specifies the SOAP body of the response to a **CreateManagedFolder** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.3.2 Elements

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

Element name	Description
CreateManagedFolder	Specifies a request to add a managed folder to a mailbox in the server message store.
CreateManagedFolderResponse	Specifies the response body content from a request to create a managed folder.

3.1.4.3.2.1 CreateManagedFolder Element

The **CreateManagedFolder** element specifies a request message for a **CreateManagedFolder** operation.

```
<xs:element name="CreateManagedFolder"
  type="m:CreateManagedFolderRequestType"
 />
```

3.1.4.3.2.2 CreateManagedFolderResponse Element

The **CreateManagedFolderResponse** element specifies a response message for a **CreateManagedFolder** operation.

```
<xs:element name="CreateManagedFolderResponse"
  type="m:CreateManagedFolderResponseType"
 />
```

3.1.4.3.3 Complex Types

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

Complex type name	Description
CreateManagedFolderRequestType	Specifies a request message for a **CreateManagedFolder** operation.
CreateManagedFolderResponseType	Specifies a response message that is returned from a **CreateManagedFolder** operation.
NonEmptyArrayOfFolderNamesType	Specifies an array of named managed folders in a mailbox. This array has at least one member.

3.1.4.3.3.1 m:CreateManagedFolderRequestType Complex Type

The **CreateManagedFolderRequestType** complex type specifies a request message to create a managed folder in a server database. The **CreateManagedFolderRequestType** complex type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.17.

```
<xs:complexType name="CreateManagedFolderRequestType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="FolderNames"
          type="t:NonEmptyArrayOfFolderNamesType"
         />
        <xs:element name="Mailbox"
          type="t:EmailAddressType"
          minOccurs="0"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
FolderNames	**t:NonEmptyArrayOfFolderNamesType** (section 3.1.4.3.3.3)	Specifies an array of managed folders to add to a mailbox. This array has at least one member.
Mailbox	**t:EmailAddressType** ([MS-OXWSCDATA] section 2.2.4.31)	Specifies the e-mail address of the mailbox in which the managed folders are added.

3.1.4.3.3.2 m:CreateManagedFolderResponseType Complex Type

The **CreateManagedFolderResponseType** complex type specifies the response message that is returned by the**CreateManagedFolder** operation. The **CreateManagedFolderResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.18.

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

3.1.4.3.3.3 t:NonEmptyArrayOfFolderNamesType Complex Type

The **NonEmptyArrayOfFolderNamesType** complex type represents an array of named managed folders in a mailbox. This array has at least one member.

```
<xs:complexType name="NonEmptyArrayOfFolderNamesType">
```

```
  <xs:sequence>
    <xs:element name="FolderName"
      type="xs:string"
      maxOccurs="unbounded"
     />
  </xs:sequence>
</xs:complexType>
```

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

Element name	Type	Description
FolderName	**xs:string** ([XMLSCHEMA2])	Specifies an array of managed folders.

3.1.4.4 DeleteFolder

The **DeleteFolder** operation deletes folders from a mailbox. The **DeleteFolder** operation is used to delete unmanaged folders and managed folders. This operation cannot delete default folders, such as the Inbox folder or the Deleted Items folder. To be deleted, a folder MUST exist.

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

```
<wsdl:operation name="DeleteFolder">
    <wsdl:input message="tns:DeleteFolderSoapIn" />
    <wsdl:output message="tns:DeleteFolderSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification of the **DeleteFolder** operation.

```
<wsdl:operation name="DeleteFolder">
    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/DeleteFolder" />
    <wsdl:input>
        <soap:header message="tns:DeleteFolderSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:DeleteFolderSoapIn" part="MailboxCulture" use="literal"/>
        <soap:header message="tns:DeleteFolderSoapIn" part="RequestVersion" use="literal"/>
        <soap:body parts="request" use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="DeleteFolderResult" use="literal" />
        <soap:header message="tns:DeleteFolderSoapOut" part="ServerVersion" use="literal"/>
    </wsdl:output>
</wsdl:operation>
```

The protocol client sends a **DeleteFolderSoapIn** request WSDL message, and the protocol server MUST respond with a **DeleteFolderSoapOut** response WSDL message.

A successful **DeleteFolder** operation request returns a **DeleteFolderResponse** element with the **ResponseClass** attribute of the **DeleteFolderResponseMessage** element set to "Success" and the **ResponseCode** element of the **DeleteFolderResponse** element set to "NoError".

An unsuccessful **DeleteFolder** operation request returns a **DeleteFolderResponse** element with the **ResponseClass** attribute of the **DeleteFolderResponseMessage** element set to "Error". The

ResponseCode element of the **DeleteFolderResponseMessage** element is set to a value of the **ResponseCodeType** simple type, as specified in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.4.1 Messages

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

Message name	Description
DeleteFolderSoapIn	Specifies the SOAP message that deletes a folder.
DeleteFolderSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.4.1.1 tns:DeleteFolderSoapIn Message

The **DeleteFolderSoapIn** WSDL message specifies the **DeleteFolder** operation request to delete a folder.

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

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

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

Part name	Element/type	Description
request	**tns:DeleteFolder** (section 3.1.4.4.2.1)	Specifies the SOAP body of the request to delete a folder.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.3.7)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the **DeleteFolder** operation request.

3.1.4.4.1.2 tns:DeleteFolderSoapOut Message

The **DeleteFolderSoapOut** WSDL message specifies the server response to the **DeleteFolder** operation request to delete a folder.

```
<wsdl:message name="DeleteFolderSoapOut">
    <wsdl:part name="DeleteFolderResult" element="tns:DeleteFolderResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
```

```
</wsdl:message>
```

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

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

Part name	Element/type	Description
DeleteFolderResult	**tns:DeleteFolderResponse** (section 3.1.4.4.2.2)	Specifies the SOAP body of the response to a **DeleteFolder** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.4.2 Elements

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

Element name	Description
DeleteFolder	Specifies a request to delete a folder from a mailbox in the server message store.
DeleteFolderResponse	Specifies the response body content from a request to delete a folder.

3.1.4.4.2.1 DeleteFolder Element

The **DeleteFolder** element specifies a request message for a **DeleteFolder** operation.

```
<xs:element name="DeleteFolder"
  type="m:DeleteFolderType"
 />
```

3.1.4.4.2.2 DeleteFolderResponse Element

The **DeleteFolderResponse** element specifies a response message for a **DeleteFolder** operation.

```
<xs:element name="DeleteFolderResponse"
  type="m:DeleteFolderResponseType"
 />
```

3.1.4.4.3 Complex Types

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

Complex type name	Description
DeleteFolderType	Specifies a request message for the **DeleteFolder** operation.

Complex type name	Description
DeleteFolderResponseType	Specifies a response message for the **DeleteFolder** operation.

3.1.4.4.3.1 m:DeleteFolderResponseType Complex Type

The **DeleteFolderResponseType** complex type specifies the response message that is returned by the**DeleteFolder** operation. The **DeleteFolderResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.18.

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

3.1.4.4.3.2 m:DeleteFolderType Complex Type

The **DeleteFolderType** complex type specifies a request message to delete folders from a mailbox. The **DeleteFolderType** complex type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.17.

```
<xs:complexType name="DeleteFolderType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="FolderIds"
          type="t:NonEmptyArrayOfBaseFolderIdsType"
         />
      </xs:sequence>
      <xs:attribute name="DeleteType"
        type="t:DisposalType"
        use="required"
       />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
FolderIds	**t:NonEmptyArrayOfBaseFolderIdsType** (section 3.1.4.6.3.3)	An array of folders to be deleted from a mailbox. This array has at least one member.

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

Attribute name	Type	Description
DeleteType	**t:DisposalType** ([MS-OXWSCDATA] section 2.2.5.9)	Describes how a folder is to be deleted.

3.1.4.5 EmptyFolder

The **EmptyFolder** operation empties identified folders<11> and can be used to delete the subfolders of the specified folder. When a subfolder is deleted, the subfolder and the messages within the subfolder are deleted.<12>

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

```
<wsdl:operation name="EmptyFolder">
    <wsdl:input message="tns:EmptyFolderSoapIn" />
    <wsdl:output message="tns:EmptyFolderSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification of the **EmptyFolder** operation.

```
<wsdl:operation name="EmptyFolder">
    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/EmptyFolder" />
    <wsdl:input>
        <soap:header message="tns:EmptyFolderSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:EmptyFolderSoapIn" part="MailboxCulture" use="literal"/>
        <soap:header message="tns:EmptyFolderSoapIn" part="RequestVersion" use="literal"/>
        <soap:body parts="request" use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="EmptyFolderResult" use="literal" />
        <soap:header message="tns:EmptyFolderSoapOut" part="ServerVersion" use="literal"/>
    </wsdl:output>
```

The protocol client sends an **EmptyFolderSoapIn** request WSDL message, and the protocol server responds with an **EmptyFolderSoapOut** response WSDL message.

A successful **EmptyFolder** operation request returns an **EmptyFolderResponse** element with the **ResponseClass** attribute of the **EmptyFolderResponseMessage** element set to "Success" and the **ResponseCode** element of the **EmptyFolderResponse** element set to "NoError".

An unsuccessful **EmptyFolder** operation request returns an **EmptyFolderResponse** element with the **ResponseClass** attribute of the **EmptyFolderResponseMessage** element set to "Error". The **ResponseCode** element of the **EmptyFolderResponseMessage** element is set to a value of the **ResponseCodeType** simple type, as specified in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.5.1 Messages

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

Message name	Description
EmptyFolderSoapIn	Specifies the SOAP message that empties a folder.

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

3.1.4.5.1.1 tns:EmptyFolderSoapIn Message

The **EmptyFolderSoapIn** WSDL message specifies the **EmptyFolder** operation request to empty a folder.

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

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

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

Part name	Element/type	Description
request	**tns:EmptyFolder** (section 3.1.4.5.2.1)	Specifies the SOAP body of the request to empty a folder.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.3.7)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the **EmptyFolder** operation request.

3.1.4.5.1.2 tns:EmptyFolderSoapOut Message

The **EmptyFolderSoapOut** WSDL message specifies the server response to the **EmptyFolder** operation request to delete a folder.

```
<wsdl:message name="EmptyFolderSoapOut">
    <wsdl:part name="EmptyFolderResult" element="tns:EmptyFolderResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

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

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

Part name	Element/type	Description
EmptyFolderResult	**tns:EmptyFolderResponse** (section 3.1.4.5.2.2)	Specifies the SOAP body of the response to a **EmptyFolder** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.5.2 Elements

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

Element name	Description
EmptyFolder	Specifies a request to empty folders in a mailbox in the server message store.
EmptyFolderResponse	Specifies the response body content from a request to empty a folder.

3.1.4.5.2.1 EmptyFolder Element

The **EmptyFolder** element specifies a request message for an **EmptyFolder** operation.

```
<xs:element name="EmptyFolder"
  type="m:EmptyFolderType"
 />
```

3.1.4.5.2.2 EmptyFolderResponse Element

The **EmptyFolderResponse** element specifies a response message for an **EmptyFolder** operation.

```
<xs:element name="EmptyFolderResponse"
  type="m:EmptyFolderResponseType"
 />
```

3.1.4.5.3 ComplexTypes

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

Complex type name	Description
EmptyFolderType	Specifies a request message for an **EmptyFolder** operation.
EmptyFolderResponseType	Specifies a response message for an **EmptyFolder** operation.

3.1.4.5.3.1 m:EmptyFolderType Complex Type

The **EmptyFolderType** complex type specifies a request message to empty a folder in a mailbox. The **EmptyFolderType** complex type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.17.

```
<xs:complexType name="EmptyFolderType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="FolderIds"
          type="t:NonEmptyArrayOfBaseFolderIdsType"
         />
      </xs:sequence>
      <xs:attribute name="DeleteType"
        type="t:DisposalType"
        use="required"
       />
      <xs:attribute name="DeleteSubFolders"
        type="xs:boolean"
        use="required"
       />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
FolderIds	**t:NonEmptyArrayOfBaseFolderIdsType** (section 3.1.4.6.3.3)	Specifies the collection of items and folders to delete.

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

Attribute name	Type	Description
DeleteType	**t:DisposalType** ([MS-OXWSCDATA] section 2.2.5.9)	Specifies an enumeration value that describes how an item is deleted.
DeleteSubFolders	**xs:boolean** ([XMLSCHEMA2])	A Boolean value that indicates whether the subfolders are also to be deleted. The **DeleteSubFolders** attribute is set to "true" if the subfolders are to be deleted; otherwise, it is set to "false".

3.1.4.5.3.2 m:EmptyFolderResponseType Complex Type

The **EmptyFolderResponseType** complex type specifies the response message that is returned by the**EmptyFolder** operation. The **EmptyFolderResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.18.

```
<xs:complexType name="EmptyFolderResponseType">
  <xs:complexContent>
```

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

3.1.4.6 GetFolder

The **GetFolder** operation gets folders, Calendar folders, Contacts folders, Tasks folders, and search folders.

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

```
<wsdl:operation name="GetFolder">
    <wsdl:input message="tns:GetFolderSoapIn" />
    <wsdl:output message="tns:GetFolderSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification of the **GetFolder** operation.

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

The protocol client sends a **GetFolderSoapIn** request WSDL message, and the protocol server responds with a **GetFolderSoapOut** response WSDL message.

A successful **GetFolder** operation request returns a **GetFolderResponse** element with the **ResponseClass** attribute of the **GetFolderResponseMessage** element set to "Success" and the **ResponseCode** element of the **GetFolderResponse** element set to "NoError".

An unsuccessful **GetFolder** operation request returns a **GetFolderResponse** element with the **ResponseClass** attribute of the **GetFolderResponseMessage** element set to "Error". The **ResponseCode** element of the **GetFolderResponseMessage** element is set to a value of the **ResponseCodeType** simple type, as specified in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.6.1 Messages

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

Message name	Description
GetFolderSoapIn	Specifies the SOAP message that gets a folder.

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

3.1.4.6.1.1 tns:GetFolderSoapIn Message

The **GetFolderSoapIn** WSDL message specifies the **GetFolder** operation request to get a folder.

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

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

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

Part name	Element/type	Description
request	**tns:GetFolder** (section 3.1.4.6.2.1)	Specifies the SOAP body of the request to get a folder.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.3.7)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the **GetFolder** operation request.
TimeZoneContext	**t:TimeZoneContext** ([MS-OXWSGTZ] section 2.2.3.4)	Specifies a time zone definition and enables the association of SOAP attributes with the definition.
ManagementRole	**t:ManagementRole** ([MS-OXWSCDATA] section 2.2.4.46)	Specifies a SOAP header that indicates which roles the caller or application wants to use.

3.1.4.6.1.2 tns:GetFolderSoapOut Message

The **GetFolderSoapOut** WSDL message specifies the server response to the **GetFolder** operation request to get a folder.

```
<wsdl:message name="GetFolderSoapOut">
    <wsdl:part name="GetFolderResult" element="tns:GetFolderResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

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

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

Part name	Element/type	Description
GetFolderResult	**tns:GetFolderResponse** (section 3.1.4.6.2.2)	Specifies the SOAP body of the response to a **GetManagedFolder** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.6.2 Elements

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

Element name	Description
GetFolder	Specifies a request to get a folder from a mailbox in the server message store.
GetFolderResponse	Specifies the response body content from a request to get a folder.

3.1.4.6.2.1 GetFolder Element

The **GetFolder** element specifies a request message for a **GetFolder** operation.

```
<xs:element name="GetFolder"
  type="m:GetFolderType"
 />
```

3.1.4.6.2.2 GetFolderResponse Element

The **GetFolderResponse** element specifies a response message for a **GetFolder** operation.

```
<xs:element name="GetFolderResponse"
  type="m:GetFolderResponseType"
 />
```

3.1.4.6.3 Complex Types

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

Complex type name	Description
GetFolderResponseType	Specifies a response message for the **GetFolder** operation.
GetFolderType	Specifies a request message for the **GetFolder** operation.
NonEmptyArrayOfBaseFolderIdsType	Specifies an array of one or more folders. This array has at least one

Complex type name	Description
	member.

3.1.4.6.3.1 m:GetFolderResponseType Complex Type

The **GetFolderResponseType** complex type specifies the response message that is returned by the **GetFolder** operation. The **GetFolderResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.18.

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

3.1.4.6.3.2 m:GetFolderType Complex Type

The **GetFolderType** complex type specifies a request message to get a folder in a server database. The **GetFolderType** complex type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.17.

```
<xs:complexType name="GetFolderType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="FolderShape"
          type="t:FolderResponseShapeType"
         />
        <xs:element name="FolderIds"
          type="t:NonEmptyArrayOfBaseFolderIdsType"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
FolderShape	**t:FolderResponseShapeType** ([MS-OXWSCDATA] section 2.2.4.37)	Specifies the properties to include in the response.
FolderIds	**t:NonEmptyArrayOfBaseFolderIdsType** (section 3.1.4.6.3.3)	An array of one or more folder identifiers. This array has at least one member.

3.1.4.6.3.3 t:NonEmptyArrayOfBaseFolderIdsType Complex Type

The **NonEmptyArrayOfBaseFolderIdsType** complex type specifies an array of one or more folder identifiers.

```
<xs:complexType name="NonEmptyArrayOfBaseFolderIdsType">
  <xs:choice
    maxOccurs="unbounded"
    minOccurs="1"
  >
    <xs:element name="FolderId"
      type="t:FolderIdType"
     />
    <xs:element name="DistinguishedFolderId"
      type="t:DistinguishedFolderIdType"
     />
  </xs:choice>
</xs:complexType>
```

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

Element name	Type	Description
FolderId	**t:FolderIdType** ([MS-OXWSCDATA] section 2.2.4.36)	Specifies the folder identifier and change key. The maximum length for the **FolderIdType** element **Id** attribute and the maximum length for the **FolerIdType ChangeKey** attribute is 512 bytes after base64 decoding.
DistinguishedFolderId	**t:DistinguishedFolderIdType** ([MS-OXWSCDATA] section 2.2.4.27)	Specifies a distinguished folder identifier.

3.1.4.7 MoveFolder

The **MoveFolder** operation moves folders from a specified parent folder and puts them in another parent folder. All the properties, contents, and subfolders of the folder move with the folder.

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

```
<wsdl:operation name="MoveFolder">
    <wsdl:input message="tns:MoveFolderSoapIn" />
    <wsdl:output message="tns:MoveFolderSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification of the **MoveFolder** operation.

```
<wsdl:operation name="MoveFolder">
    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/MoveFolder" />
    <wsdl:input>
        <soap:header message="tns:MoveFolderSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:MoveFolderSoapIn" part="MailboxCulture" use="literal"/>
        <soap:header message="tns:MoveFolderSoapIn" part="RequestVersion" use="literal"/>
        <soap:body parts="request" use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="MoveFolderResult" use="literal" />

        <soap:header message="tns:MoveFolderSoapOut" part="ServerVersion" use="literal"/>
```

```
    </wsdl:output>
</wsdl:operation>
```

The protocol client sends a **MoveFolderSoapIn** request WSDL message, and the protocol server responds with a **MoveFolderSoapOut** response WSDL message.

A successful **MoveFolder** operation request returns a **MoveFolderResponse** element with the **ResponseClass** attribute of the **MoveFolderResponseMessage** element set to "Success" and the **ResponseCode** element of the **MoveFolderResponse** element set to "NoError".

An unsuccessful **MoveFolder** operation request returns a **MoveFolderResponse** element with the **ResponseClass** attribute of the **MoveFolderResponseMessage** element set to "Error". The **ResponseCode** element of the **MoveFolderResponseMessage** element is set to a value of the **ResponseCodeType** simple type, as specified in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.7.1 Messages

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

Message name	Description
MoveFolderSoapIn	Specifies the SOAP message that moves a folder.
MoveFolderSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.7.1.1 tns:MoveFolderSoapIn Message

The **MoveFolderSoapIn** WSDL message specifies the **MoveFolder** operation request to move a folder.

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

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

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

Part name	Element/type	Description
request	**tns:MoveFolder** (section 3.1.4.7.2.1)	Specifies the SOAP body of the request to move a folder.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.3.7)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-	Specifies a SOAP header that identifies the

Part name	Element/type	Description
	OXWSCDATA] section 2.2.3.11)	schema version for the **MoveFolder** operation request.

3.1.4.7.1.2 tns:MoveFolderSoapOut Message

The **MoveFolderSoapOut** WSDL message specifies the server response to the **MoveFolder** operation request to the move a folder.

```
<wsdl:message name="MoveFolderSoapOut">
    <wsdl:part name="MoveFolderResult" element="tns:MoveFolderResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

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

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

Part name	Element/type	Description
MoveFolderResult	**tns:MoveFolderResponse** (section 3.1.4.7.2.2)	Specifies the SOAP body of the response to a **MoveFolder** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.7.2 Elements

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

Element name	Description
MoveFolder	Specifies a request to move a folder in the server message store.
MoveFolderResponse	Specifies the response body content from a request to move a folder.

3.1.4.7.2.1 MoveFolder Element

The **MoveFolder** element specifies a request message for a **MoveFolder** operation.

```
<xs:element name="MoveFolder"
  type="m:MoveFolderType"
/>
```

3.1.4.7.2.2 MoveFolderResponse Element

The **MoveFolderResponse** element specifies a response message for a **MoveFolder** operation.

```
<xs:element name="MoveFolderResponse"
```

```
 type="m:MoveFolderResponseType"
/>
```

3.1.4.7.3 Complex Types

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

Complex type name	Description
MoveFolderResponseType	Specifies a response message for the **MoveFolder** operation.
MoveFolderType	Specifies a request message for the **MoveFolder** operation

3.1.4.7.3.1 m:MoveFolderResponseType Complex Type

The **MoveFolderResponseType** complex type specifies the response message that is returned by the **MoveFolder** operation. The **MoveFolderResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.18.

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

3.1.4.7.3.2 m:MoveFolderType Complex Type

The **MoveFolderType** complex type specifies a request message to move folders in a mailbox. The **MoveFolderType** complex type extends the **BaseMoveCopyFolderType,** as specified in section 2.2.4.6.

```
<xs:complexType name="MoveFolderType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseMoveCopyFolderType"
     />
  </xs:complexContent>
</xs:complexType>
```

3.1.4.8 UpdateFolder

The **UpdateFolder** operation modifies properties of an existing folder in the server message store.

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

```
<wsdl:operation name="UpdateFolder">
    <wsdl:input message="tns:UpdateFolderSoapIn" />
    <wsdl:output message="tns:UpdateFolderSoapOut" />
```

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

The following is the WSDL binding specification of the **UpdateFolder** operation.

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

The protocol client sends an **UpdateFolderSoapIn** request WSDL message, and the protocol server responds with a **UpdateFolderSoapOut** response WSDL message.

A successful **UpdateFolder** operation request returns an **UpdateFolderResponse** element with the **ResponseClass** attribute of the **UpdateFolderResponseMessage** element set to "Success" and the **ResponseCode** element of the **UpdateFolderResponse** element set to "NoError".

An unsuccessful **UpdateFolder** operation request returns an **UpdateFolderResponse** element with the **ResponseClass** attribute of the **UpdateFolderResponseMessage** element set to "Error". The **ResponseCode** element of the **UpdateFolderResponseMessage** element is set to a value of the **ResponseCodeType** simple type, as specified in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.8.1 Messages

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

Message name	Description
UpdateFolderSoapIn	Specifies the SOAP message that updates a folder.
UpdateFolderSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.8.1.1 tns:UpdateFolderSoapIn Message

The **UpdateFolderSoapIn** WSDL message specifies the **UpdateFolder** operation request to update a folder.

```
<wsdl:message name="UpdateFolderSoapIn">
    <wsdl:part name="request" element="tns:UpdateFolder" />
    <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>
```

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

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

Part Name	Element/type	Description
request	**tns:UpdateFolder** (section 3.1.4.8.2.1)	Specifies the SOAP body of the request to create a folder.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.3.7)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the **UpdateFolder** operation request.
TimeZoneContext	**t:TimeZoneContext** ([MS-OXWSGTZ] section 2.2.3.4)	Specifies a time zone definition and enables the association of SOAP attributes with the definition.

3.1.4.8.1.2 tns:UpdateFolderSoapOut Message

The **UpdateFolderSoapOut** WSDL message specifies the server response to the **UpdateFolder** operation request to update a folder.

```
<wsdl:message name="UpdateFolderSoapOut">
    <wsdl:part name="UpdateFolderResult" element="tns:UpdateFolderResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

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

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

Part name	Element/type	Description
UpdateFolderResult	**tns:UpdateFolderResponse** (section 3.1.4.8.2.2)	Specifies the SOAP body of the response to a **UpdateFolder** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.8.2 Elements

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

Element name	Description
UpdateFolder	Specifies a request to update a folder in the server message store.
UpdateFolderResponse	Specifies the response body content from a request to update a folder.

3.1.4.8.2.1 UpdateFolder Element

The **UpdateFolder** element specifies a request message for an **UpdateFolder** operation.

```
<xs:element name="UpdateFolder"
  type="m:UpdateFolderType"
 />
```

3.1.4.8.2.2 UpdateFolderResponse Element

The **UpdateFolderResponse** element specifies a response message for an **UpdateFolder** operation request.

```
<xs:element name="UpdateFolderResponse"
  type="m:UpdateFolderResponseType"
 />
```

3.1.4.8.3 Complex Types

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

Complex type name	Description
UpdateFolderResponseType	Specifies a response message for the **UpdateFolder** operation.
UpdateFolderType	Specifies a request message for the **UpdateFolder** operation.
AppendToFolderFieldType	This complex type is not implemented.
DeleteFolderFieldType	Specifies an **UpdateFolder** operation to delete a property from a folder.
NonEmptyArrayOfFolderChangeDescriptionsType	Specifies an array of **ChangeDescriptionType** complex types (section 3.1.4.8.3.5). This array has at least one member.
NonEmptyArrayOfFolderChangesType	Specifies an array of **FolderChangeType** complex types (section 3.1.4.8.3.6). This array has at least one member.
SetFolderFieldType	Specifies an **UpdateFolder** operation to set a property on an existing folder.

3.1.4.8.3.1 m:UpdateFolderResponseType Complex Type

The **UpdateFolderResponseType** complex type specifies the response message that is returned by the **UpdateFolder** operation. The **UpdateFolderResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.18.

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

3.1.4.8.3.2 m:UpdateFolderType Complex Type

The **UpdateFolderType** complex type specifies a request message to update folders in a mailbox. The **UpdateFolderType** complex type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.17.

```
<xs:complexType name="UpdateFolderType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="FolderChanges"
          type="t:NonEmptyArrayOfFolderChangesType"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
FolderChanges	**t:NonEmptyArrayOfFolderChangesType** (section 3.1.4.8.3.6)	Represents an array of folders to be updated.

3.1.4.8.3.3 t:AppendToFolderFieldType Complex Type

The **AppendToFolderFieldType** complex type is not implemented. Any request that uses this complex type will always return an error response. The **AppendToFolderFieldType** complex type extends the **FolderChangeDescriptionType** complex type, as specified in section 2.2.4.9.

```
<xs:complexType name="AppendToFolderFieldType">
  <xs:complexContent>
    <xs:extension
      base="t:FolderChangeDescriptionType"
    >
      <xs:sequence>
        <xs:choice>
          <xs:element name="Folder"
            type="t:FolderType"
           />
          <xs:element name="CalendarFolder"
```

```
            type="t:CalendarFolderType"
           />
          <xs:element name="ContactsFolder"
            type="t:ContactsFolderType"
           />
          <xs:element name="SearchFolder"
            type="t:SearchFolderType"
           />
          <xs:element name="TasksFolder"
            type="t:TasksFolderType"
           />
        </xs:choice>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
Folder	**t:FolderType** (section 2.2.4.12)	Represents a regular folder in the server database.
CalendarFolder	**t:CalendarFolderType** ([MS-OXWSMTGS] section 2.2.4.9)	Represents a folder that primarily contains calendar items.
ContactsFolder	**t:ContactsFolderType** ([MS-OXWSCONT] section 3.1.4.1.1.6)	Represents a Contacts folder in a mailbox.
SearchFolder	**t:SearchFolderType** ([MS-OXWSSRCH] section 2.2.4.26)	Represents a search folder that is contained in a mailbox.
TasksFolder	**t:TasksFolderType** ([MS-OXWSTASK] section 2.2.4.5)	Represents a Tasks folder that is contained in a mailbox.

3.1.4.8.3.4 t:DeleteFolderFieldType Complex Type

The **DeleteFolderFieldType** complex type represents an **UpdateFolder** operation to delete a property from a folder. The **DeleteFolderFieldType** complex type extends the **FolderChangeDescriptionType** complex type, as specified in section 2.2.4.9.

```
<xs:complexType name="DeleteFolderFieldType">
  <xs:complexContent>
    <xs:extension
      base="t:FolderChangeDescriptionType"
    >
      <xs:sequence />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

3.1.4.8.3.5 t:NonEmptyArrayOfFolderChangeDescriptionsType Complex Type

The **NonEmptyArrayOfFolderChangeDescriptionsType** complex type represents an array of **ChangeDescriptionType** complex types, as specified in [MS-OXWSCORE] section 3.1.4.9.3.4. This array has at least one member.

```
<xs:complexType name="NonEmptyArrayOfFolderChangeDescriptionsType">
  <xs:choice
    maxOccurs="unbounded"
  >
    <xs:element name="AppendToFolderField"
      type="t:AppendToFolderFieldType"
     />
    <xs:element name="SetFolderField"
      type="t:SetFolderFieldType"
     />
    <xs:element name="DeleteFolderField"
      type="t:DeleteFolderFieldType"
     />
  </xs:choice>
</xs:complexType>
```

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

Element name	Type	Description
AppendToFolderField	**t:AppendToFolderFieldType**(section 3.1.4.8.3.3)	This complex type is not implemented.
SetFolderField	**t:SetFolderFieldType** (section 3.1.4.8.3.7)	Represents an **UpdateFolder** operation to set a property on an existing folder.
DeleteFolderField	**t:DeleteFolderFieldType** (section 3.1.4.8.3.4)	Represents an **UpdateFolder** operation to delete a property from a folder.

The **FolderChangeDescriptionType** complex type, as specified in section 2.2.4.9, members describe a change to a single folder property.

3.1.4.8.3.6 t:NonEmptyArrayOfFolderChangesType Complex Type

The **NonEmptyArrayOfFolderChangesType** complex type represents an array of **FolderChangeType** complex types, as specified in section 2.2.4.10. This array has at least one member.

```
<xs:complexType name="NonEmptyArrayOfFolderChangesType">
  <xs:sequence>
    <xs:element name="FolderChange"
      type="t:FolderChangeType"
      maxOccurs="unbounded"
     />
  </xs:sequence>
</xs:complexType>
```

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

Element name	Type	Description
FolderChange	**t:FolderChangeType** (section 2.2.4.10)	Represents a collection of changes to be performed on a single folder.

3.1.4.8.3.7 t:SetFolderFieldType Complex Type

The **SetFolderFieldType** complex type represents an **UpdateFolder** operation to set a property on an existing folder. The **SetFolderFieldType** complex type extends the **FolderChangeDescriptionType** complex type, as specified in section 2.2.4.9.

```
<xs:complexType name="SetFolderFieldType">
  <xs:complexContent>
    <xs:extension
      base="t:FolderChangeDescriptionType"
    >
      <xs:choice>
        <xs:element name="Folder"
          type="t:FolderType"
         />
        <xs:element name="CalendarFolder"
          type="t:CalendarFolderType"
         />
        <xs:element name="ContactsFolder"
          type="t:ContactsFolderType"
         />
        <xs:element name="SearchFolder"
          type="t:SearchFolderType"
         />
        <xs:element name="TasksFolder"
          type="t:TasksFolderType"
         />
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
Folder	**t:FolderType** (section 2.2.4.12)	Represents a regular folder in the server database.
CalendarFolder	**t:CalendarFolderType** ([MS-OXWSMTGS] section 2.2.4.9)	Represents a folder that primarily contains calendar items.
ContactsFolder	**t:ContactsFolderType** ([MS-OXWSCONT] section 3.1.4.1.1.6)	Represents a Contacts folder in a mailbox.
SearchFolder	**t:SearchFolderType** ([MS-OXWSSRCH] section 2.2.4.26)	Represents a search folder that is contained in a mailbox.
TasksFolder	**t:TasksFolderType** ([MS-OXWSTASK] section 2.2.4.5)	Represents a Tasks folder that is contained in a mailbox.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following examples show the request and response XML to perform the specific operations.

4.1 CopyFolder Operation

The following is an example of a **CopyFolder** operation with the **CopyFolderType** complex type.

The client constructs the request XML and sends it to the server. The identified folder, along with all contents, is copied to the destination folder, in this example, the **Junk E-mail folder**. The **FolderId Id** and **ChangeKey** attributes have been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8" ?>
  <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
        xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header>
      <t:RequestServerVersion Version="Exchange2010" />
    </soap:Header>
    <soap:Body>
      <m:CopyFolder>
        <m:ToFolderId>
          <t:DistinguishedFolderId Id="junkemail" />
        </m:ToFolderId>
        <m:FolderIds>
          <t:FolderId Id="AAMkAGIwODEy" ChangeKey="AQAAABYAAA" />
        </m:FolderIds>
      </m:CopyFolder>
    </soap:Body>
  </soap:Envelope>
```

The server constructs the response XML and sends it to the client. The **FolderId** element **Id** and **ChangeKey** attributes have been shortened to preserve readability.

```
  <?xml version="1.0" encoding="utf-8"?>
  <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Header>
      <h:ServerVersionInfo MajorVersion="14"
          MinorVersion="1"
          MajorBuildNumber="63"
          MinorBuildNumber="0"
          Version="Exchange2010"
          xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
          xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
    </s:Header>
    <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <m:CopyFolderResponse
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
          xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
        <m:ResponseMessages>
          <m:CopyFolderResponseMessage ResponseClass="Success">
            <m:ResponseCode>NoError</m:ResponseCode>
            <m:Folders>
              <t:Folder>
                <t:FolderId Id="AAMkAGIwODEy" ChangeKey="AQAAABYAAA" />
              </t:Folder>
            </m:Folders>
          </m:CopyFolderResponseMessage>
        </m:ResponseMessages>
```

```
      </m:CopyFolderResponse>
    </s:Body>
  </s:Envelope>
```

4.2 CreateFolder Operation

The following is an example of a **CreateFolder** operation with the **CreateFolderType** complex type.

The client constructs the request XML and sends it to the server. A new folder named Custom Folder is created in the Inbox. If the newly created folder already exists within the specified parent folder, an error is thrown.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010" />
  </soap:Header>
  <soap:Body>
    <m:CreateFolder>
      <m:ParentFolderId>
        <t:DistinguishedFolderId Id="inbox" />
      </m:ParentFolderId>
      <m:Folders>
        <t:Folder>
          <t:FolderClass>IPF.MyCustomFolderClass</t:FolderClass>
          <t:DisplayName>Custom Folder</t:DisplayName>
          <t:PermissionSet>
            <t:Permissions />
          </t:PermissionSet>
        </t:Folder>
      </m:Folders>
    </m:CreateFolder>
  </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client. The **FolderId** element **Id** and **ChangeKey** attributes have been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
       MinorVersion="1"
       MajorBuildNumber="63"
       MinorBuildNumber="0"
       Version="Exchange2010"
       xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
       xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <m:CreateFolderResponse
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
       xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:CreateFolderResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
          <m:Folders>
```

```
            <t:Folder>
              <t:FolderId Id="AAMkAGIwODEy" ChangeKey="AQAAABYAAA " />
            </t:Folder>
          </m:Folders>
        </m:CreateFolderResponseMessage>
      </m:ResponseMessages>
    </m:CreateFolderResponse>
  </s:Body>
</s:Envelope>
```

4.3 DeleteFolder Operation

The following is an example of a **DeleteFolder** operation with the **DeleteFolderType** complex type.

The client constructs the request XML and sends it to the server. The **FolderId** element **Id** and **ChangeKey** attributes have been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010" />
  </soap:Header>
  <soap:Body>
    <m:DeleteFolder DeleteType="SoftDelete">
      <m:FolderIds>
        <t:FolderId Id="AAMkAGIwODEy" ChangeKey="AQAAABQAAAA" />
      </m:FolderIds>
    </m:DeleteFolder>
  </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
        MinorVersion="1"
        MajorBuildNumber="63"
        MinorBuildNumber="0"
        Version="Exchange2010"
        xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <m:DeleteFolderResponse
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
        xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:DeleteFolderResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
        </m:DeleteFolderResponseMessage>
      </m:ResponseMessages>
    </m:DeleteFolderResponse>
  </s:Body>
</s:Envelope>
```

4.4 EmptyFolder Operation

The following is an example of an **EmptyFolder** operation with the **EmptyFolderType** complex type.

The client constructs the request XML and sends it to the server. This example of an **EmptyFolder** operation request shows how to form a request to empty a folder. This example deletes all subfolders of the identified folder.

The **FolderId** element **Id** and **ChangeKey** attributes have been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
               xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
  <soap:Header>
    <t:RequestServerVersion Version ="Exchange2010_SP1"/>
    </soap:Header>
    <soap:Body>
      <m:EmptyFolder DeleteType="HardDelete" DeleteSubFolders="true">
        <m:FolderIds>
          <t:FolderId Id="AQMkADhhOGU0"  ChangeKey="AQAAABYAAABsMB" />
        </m:FolderIds>
      </m:EmptyFolder>
    </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
                         MinorVersion="1"
                         MajorBuildNumber="164"
                         MinorBuildNumber="0"
                         Version="Exchange2010_SP1"
                         xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
                         xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
                         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                         xmlns:xsd="http://www.w3.org/2001/XMLSchema"/>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <m:EmptyFolderResponse
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"

xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:EmptyFolderResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
        </m:EmptyFolderResponseMessage>
      </m:ResponseMessages>
    </m:EmptyFolderResponse>
  </s:Body>
</s:Envelope>
```

4.5 MoveFolder Operation

The following is an example of a **MoveFolder** operation with the **MoveFolderType** complex type. The specified folder is from its current parent folder to the **Sent Items folder**.

The client constructs the request XML and sends it to the server. The specified folder is moved to the specified parent folder. The **FolderId** element **Id** and **ChangeKey** attributes have been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010" />
  </soap:Header>
  <soap:Body>
    <m:MoveFolder>
      <m:ToFolderId>
        <t:DistinguishedFolderId Id="sentitems" />
      </m:ToFolderId>
      <m:FolderIds>
        <t:FolderId Id="AAMkAGIwODEy" ChangeKey="AQAAABYAAA " />
      </m:FolderIds>
    </m:MoveFolder>
  </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client. The **FolderId** element **Id** and **ChangeKey** attributes have been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
      MinorVersion="1"
      MajorBuildNumber="63"
      MinorBuildNumber="0"
      Version="Exchange2010"
      xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <m:MoveFolderResponse
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
        xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:MoveFolderResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
          <m:Folders>
            <t:Folder>
              <t:FolderId Id="AAMkAGIwODEy" ChangeKey="AQAAABYAAA " />
            </t:Folder>
          </m:Folders>
        </m:MoveFolderResponseMessage>
      </m:ResponseMessages>
    </m:MoveFolderResponse>
  </s:Body>
</s:Envelope>
```

4.6 UpdateFolder Operation

The following is an example of an **UpdateFolder** operation with the **UpdateFolderType** complex type. This example modifies the display name of the identified folder to "Modified Custom Folder".

The client constructs the XML and sends it to the server. The **FolderId** element **Id** and **ChangeKey** attributes have been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
        xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010" />
  </soap:Header>
  <soap:Body>
    <m:UpdateFolder>
      <m:FolderChanges>
        <t:FolderChange>
          <t:FolderId Id="AAMkAGIwODEy" ChangeKey="AQAAABQAAAA" />
          <t:Updates>
            <t:SetFolderField>
              <t:FieldURI FieldURI="folder:DisplayName" />
              <t:Folder>
                <t:DisplayName>Modified Custom Folder</t:DisplayName>
              </t:Folder>
            </t:SetFolderField>
          </t:Updates>
        </t:FolderChange>
      </m:FolderChanges>
    </m:UpdateFolder>
  </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client. The **FolderId** element **Id** and **ChangeKey** attributes have been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
        MinorVersion="1"
        MajorBuildNumber="63"
        MinorBuildNumber="0"
        Version="Exchange2010"
        xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <m:UpdateFolderResponse
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
          xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:UpdateFolderResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
          <m:Folders>
            <t:Folder>
              <t:FolderId Id="AAMkAGIwODEy" ChangeKey="AQAAABYAAA" />
            </t:Folder>
          </m:Folders>
        </m:UpdateFolderResponseMessage>
      </m:ResponseMessages>
    </m:UpdateFolderResponse>
  </s:Body>
</s:Envelope>ap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

The XML files that are listed in the following table are required in order to implement the functionality specified in this document.

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

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

This section contains the contents of the MS-OXWSFOLD.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="Exchange2016"
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:include schemaLocation="MS-OXWSFOLD-messages.xsd"/>
            <!-- Add global elements and types from messages.xsd -->
        </xs:schema>
        <xs:schema id="types" elementFormDefault="qualified" version="Exchange2016"
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="CreateFolder">
            <wsdl:input message="tns:CreateFolderSoapIn" />
            <wsdl:output message="tns:CreateFolderSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="DeleteFolder">
            <wsdl:input message="tns:DeleteFolderSoapIn" />
            <wsdl:output message="tns:DeleteFolderSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="EmptyFolder">
          <wsdl:input message="tns:EmptyFolderSoapIn" />
          <wsdl:output message="tns:EmptyFolderSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="UpdateFolder">
            <wsdl:input message="tns:UpdateFolderSoapIn" />
            <wsdl:output message="tns:UpdateFolderSoapOut" />
        </wsdl:operation>
```

```
        <wsdl:operation name="MoveFolder">
            <wsdl:input message="tns:MoveFolderSoapIn" />
            <wsdl:output message="tns:MoveFolderSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="CopyFolder">
            <wsdl:input message="tns:CopyFolderSoapIn" />
            <wsdl:output message="tns:CopyFolderSoapOut" />
        </wsdl:operation>
                <wsdl:operation name="GetFolder">
            <wsdl:input message="tns:GetFolderSoapIn" />
            <wsdl:output message="tns:GetFolderSoapOut" />
        </wsdl:operation>
                <wsdl:operation name="CreateManagedFolder">
            <wsdl:input message="tns:CreateManagedFolderSoapIn" />
            <wsdl:output message="tns:CreateManagedFolderSoapOut" />
        </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="GetFolder">
            <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetFolder" />
            <wsdl:input>
                <soap:header message="tns:GetFolderSoapIn" part="Impersonation"
use="literal"/>
                <soap:header message="tns:GetFolderSoapIn" part="MailboxCulture"
use="literal"/>
                <soap:header message="tns:GetFolderSoapIn" part="RequestVersion"
use="literal"/>
                <soap:header message="tns:GetFolderSoapIn" part="TimeZoneContext"
use="literal"/>
                <soap:header message="tns:GetFolderSoapIn" part="ManagementRole"
use="literal"/>
                <soap:body parts="request" use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body parts="GetFolderResult" use="literal" />
                <soap:header message="tns:GetFolderSoapOut" part="ServerVersion"
use="literal"/>
            </wsdl:output>
        </wsdl:operation>
              <wsdl:operation name="CreateFolder">
            <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CreateFolder" />
            <wsdl:input>
                <soap:header message="tns:CreateFolderSoapIn" part="Impersonation"
use="literal"/>
                <soap:header message="tns:CreateFolderSoapIn" part="MailboxCulture"
use="literal"/>
                <soap:header message="tns:CreateFolderSoapIn" part="RequestVersion"
use="literal"/>
                <soap:header message="tns:CreateFolderSoapIn" part="TimeZoneContext"
use="literal"/>
                <soap:body parts="request" use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body parts="CreateFolderResult" use="literal" />
                <soap:header message="tns:CreateFolderSoapOut" part="ServerVersion"
use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="DeleteFolder">
            <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/DeleteFolder" />
```

```
            <wsdl:input>
                <soap:header message="tns:DeleteFolderSoapIn" part="Impersonation"
use="literal"/>
                <soap:header message="tns:DeleteFolderSoapIn" part="MailboxCulture"
use="literal"/>
                <soap:header message="tns:DeleteFolderSoapIn" part="RequestVersion"
use="literal"/>
                <soap:body parts="request" use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body parts="DeleteFolderResult" use="literal" />
                <soap:header message="tns:DeleteFolderSoapOut" part="ServerVersion"
use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="EmptyFolder">
          <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/EmptyFolder" />
          <wsdl:input>
            <soap:header message="tns:EmptyFolderSoapIn" part="Impersonation" use="literal"/>
            <soap:header message="tns:EmptyFolderSoapIn" part="MailboxCulture"
use="literal"/>
            <soap:header message="tns:EmptyFolderSoapIn" part="RequestVersion"
use="literal"/>
            <soap:body parts="request" use="literal" />
          </wsdl:input>
          <wsdl:output>
            <soap:body parts="EmptyFolderResult" use="literal" />
            <soap:header message="tns:EmptyFolderSoapOut" part="ServerVersion"
use="literal"/>
          </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="UpdateFolder">
            <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/UpdateFolder" />
            <wsdl:input>
                <soap:header message="tns:UpdateFolderSoapIn" part="Impersonation"
use="literal"/>
                <soap:header message="tns:UpdateFolderSoapIn" part="MailboxCulture"
use="literal"/>
                <soap:header message="tns:UpdateFolderSoapIn" part="RequestVersion"
use="literal"/>
                <soap:header message="tns:UpdateFolderSoapIn" part="TimeZoneContext"
use="literal"/>
                <soap:body parts="request" use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body parts="UpdateFolderResult" use="literal" />
                <soap:header message="tns:UpdateFolderSoapOut" part="ServerVersion"
use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="MoveFolder">
            <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/MoveFolder" />
            <wsdl:input>
                <soap:header message="tns:MoveFolderSoapIn" part="Impersonation"
use="literal"/>
                <soap:header message="tns:MoveFolderSoapIn" part="MailboxCulture"
use="literal"/>
                <soap:header message="tns:MoveFolderSoapIn" part="RequestVersion"
use="literal"/>
                <soap:body parts="request" use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body parts="MoveFolderResult" use="literal" />
                <soap:header message="tns:MoveFolderSoapOut" part="ServerVersion"
use="literal"/>
            </wsdl:output>
```

```
        </wsdl:operation>
        <wsdl:operation name="CopyFolder">
            <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CopyFolder" />
            <wsdl:input>
                <soap:header message="tns:CopyFolderSoapIn" part="Impersonation"
use="literal"/>
                <soap:header message="tns:CopyFolderSoapIn" part="MailboxCulture"
use="literal"/>
                <soap:header message="tns:CopyFolderSoapIn" part="RequestVersion"
use="literal"/>
                <soap:body parts="request" use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body parts="CopyFolderResult" use="literal" />
                <soap:header message="tns:CopyFolderSoapOut" part="ServerVersion"
use="literal"/>
            </wsdl:output>
        </wsdl:operation>
               <wsdl:operation name="CreateManagedFolder">
            <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CreateManagedFolder"
/>
            <wsdl:input>
                <soap:header message="tns:CreateManagedFolderSoapIn" part="Impersonation"
use="literal"/>
                <soap:header message="tns:CreateManagedFolderSoapIn" part="MailboxCulture"
use="literal"/>
                <soap:header message="tns:CreateManagedFolderSoapIn" part="RequestVersion"
use="literal"/>
                <soap:body parts="request" use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body parts="CreateManagedFolderResult" use="literal" />
                <soap:header message="tns:CreateManagedFolderSoapOut" part="ServerVersion"
use="literal"/>
            </wsdl:output>
        </wsdl:operation>


    </wsdl:binding>
    <wsdl:message name="GetFolderSoapIn">
        <wsdl:part name="request" element="tns:GetFolder" />
        <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
        <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
        <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
        <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext"/>
        <wsdl:part name="ManagementRole" element="t:ManagementRole"/>
    </wsdl:message>
    <wsdl:message name="GetFolderSoapOut">
        <wsdl:part name="GetFolderResult" element="tns:GetFolderResponse" />
        <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
    </wsdl:message>
    <wsdl:message name="CreateFolderSoapIn">
        <wsdl:part name="request" element="tns:CreateFolder" />
        <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="CreateFolderSoapOut">
        <wsdl:part name="CreateFolderResult" element="tns:CreateFolderResponse" />
        <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
    </wsdl:message>
    <wsdl:message name="CreateManagedFolderSoapIn">
        <wsdl:part name="request" element="tns:CreateManagedFolder" />
        <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
        <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
        <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
```

```
    </wsdl:message>
    <wsdl:message name="CreateManagedFolderSoapOut">
        <wsdl:part name="CreateManagedFolderResult" element="tns:CreateManagedFolderResponse"
/>
        <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
    </wsdl:message>
    <wsdl:message name="DeleteFolderSoapIn">
        <wsdl:part name="request" element="tns:DeleteFolder" />
        <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
        <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
        <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
    </wsdl:message>
    <wsdl:message name="DeleteFolderSoapOut">
        <wsdl:part name="DeleteFolderResult" element="tns:DeleteFolderResponse" />
        <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
    </wsdl:message>
    <wsdl:message name="EmptyFolderSoapIn">
      <wsdl:part name="request" element="tns:EmptyFolder" />
      <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
      <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
      <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
    </wsdl:message>
    <wsdl:message name="EmptyFolderSoapOut">
      <wsdl:part name="EmptyFolderResult" element="tns:EmptyFolderResponse" />
      <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
    </wsdl:message>
    <wsdl:message name="UpdateFolderSoapIn">
        <wsdl:part name="request" element="tns:UpdateFolder" />
        <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="UpdateFolderSoapOut">
        <wsdl:part name="UpdateFolderResult" element="tns:UpdateFolderResponse" />
        <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
    </wsdl:message>
    <wsdl:message name="MoveFolderSoapIn">
        <wsdl:part name="request" element="tns:MoveFolder" />
        <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
        <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
        <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
    </wsdl:message>
    <wsdl:message name="MoveFolderSoapOut">
        <wsdl:part name="MoveFolderResult" element="tns:MoveFolderResponse" />
        <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
    </wsdl:message>
    <wsdl:message name="CopyFolderSoapIn">
        <wsdl:part name="request" element="tns:CopyFolder" />
        <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
        <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
        <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
    </wsdl:message>
    <wsdl:message name="CopyFolderSoapOut">
        <wsdl:part name="CopyFolderResult" element="tns:CopyFolderResponse" />
        <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
    </wsdl:message>
</wsdl:definitions>
```

7 Appendix B: Full XML Schema

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

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

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

7.1 Messages Schema

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

MS-OXWSFOLD-messages.xsd includes the files listed in the following table. For the schema file to operate correctly, these files have to be placed 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 7.1
MS-OXWSFOLD-types.xsd	7.2

```
<?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="Exchange2016" id="messages">

  <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="MS-OXWSFOLD-TYPES.xsd"/>
  <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>


  <xs:complexType name="BaseMoveCopyFolderType">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:element name="ToFolderId" type="t:TargetFolderIdType"/>
          <xs:element name="FolderIds" type="t:NonEmptyArrayOfBaseFolderIdsType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="CopyFolder" type="m:CopyFolderType"/>
  <xs:complexType name="CopyFolderType">
    <xs:complexContent>
      <xs:extension base="m:BaseMoveCopyFolderType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="CopyFolderResponse" type="m:CopyFolderResponseType"/>
  <xs:complexType name="CopyFolderResponseType">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
```

```
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="CreateFolderResponseType">
  <xs:complexContent>
    <xs:extension base="m:BaseResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
<xs:element name="CreateFolderResponse" type="m:CreateFolderResponseType"/>
<xs:complexType name="CreateFolderType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:sequence>
        <xs:element name="ParentFolderId" type="t:TargetFolderIdType"/>
        <xs:element name="Folders" type="t:NonEmptyArrayOfFoldersType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="CreateFolder" type="m:CreateFolderType"/>
<xs:complexType name="CreateManagedFolderRequestType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:sequence>
        <xs:element name="FolderNames" type="t:NonEmptyArrayOfFolderNamesType"/>
        <xs:element name="Mailbox" type="t:EmailAddressType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="CreateManagedFolder" type="m:CreateManagedFolderRequestType"/>
<xs:complexType name="CreateManagedFolderResponseType">
  <xs:complexContent>
    <xs:extension base="m:BaseResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
<xs:element name="CreateManagedFolderResponse" type="m:CreateManagedFolderResponseType"/>
<xs:complexType name="DeleteFolderType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:sequence>
        <xs:element name="FolderIds" type="t:NonEmptyArrayOfBaseFolderIdsType"/>
      </xs:sequence>
      <xs:attribute name="DeleteType" type="t:DisposalType" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="DeleteFolder" type="m:DeleteFolderType"/>
<xs:complexType name="DeleteFolderResponseType">
  <xs:complexContent>
    <xs:extension base="m:BaseResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
<xs:element name="DeleteFolderResponse" type="m:DeleteFolderResponseType"/>
<xs:complexType name="EmptyFolderType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:sequence>
        <xs:element name="FolderIds" type="t:NonEmptyArrayOfBaseFolderIdsType"/>
      </xs:sequence>
      <xs:attribute name="DeleteType" type="t:DisposalType" use="required"/>
      <xs:attribute name="DeleteSubFolders" type="xs:boolean" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="EmptyFolder" type="m:EmptyFolderType"/>
<xs:complexType name="EmptyFolderResponseType">
  <xs:complexContent>
    <xs:extension base="m:BaseResponseMessageType"/>
  </xs:complexContent>
```

```
  </xs:complexType>
  <xs:element name="EmptyFolderResponse" type="m:EmptyFolderResponseType"/>
  <xs:complexType name="FolderInfoResponseMessageType">
    <xs:complexContent>
      <xs:extension base="m:ResponseMessageType">
        <xs:sequence>
          <xs:element name="Folders" type="t:ArrayOfFoldersType" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="GetFolderType">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:element name="FolderShape" type="t:FolderResponseShapeType"/>
          <xs:element name="FolderIds" type="t:NonEmptyArrayOfBaseFolderIdsType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="GetFolder" type="m:GetFolderType"/>
  <xs:complexType name="GetFolderResponseType">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="GetFolderResponse" type="m:GetFolderResponseType"/>
  <xs:complexType name="MoveFolderResponseType">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="MoveFolderResponse" type="m:MoveFolderResponseType"/>
  <xs:complexType name="MoveFolderType">
    <xs:complexContent>
      <xs:extension base="m:BaseMoveCopyFolderType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="MoveFolder" type="m:MoveFolderType"/>
  <xs:complexType name="UpdateFolderType">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:element name="FolderChanges" type="t:NonEmptyArrayOfFolderChangesType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="UpdateFolder" type="m:UpdateFolderType"/>
  <xs:complexType name="UpdateFolderResponseType">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="UpdateFolderResponse" type="m:UpdateFolderResponseType"/>
</xs:schema>
```

7.2 Types Schema

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

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

File name	Defining specification
MS-OXWSCDATA-types.xsd	[MS-OXWSCDATA] section 7.2
MS-OXWSXPROP-types.xsd	[MS-OXWSXPROP] section 5

```
<?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="Exchange2016" id="types">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSXPROP-types.xsd"/>
  <xs:complexType name="AddressListIdType">
    <xs:annotation>
      <xs:documentation>Identifier for a address list</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="t:BaseFolderIdType">
        <xs:attribute name="Id" type="xs:string" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="AppendToFolderFieldType">
    <xs:complexContent>
      <xs:extension base="t:FolderChangeDescriptionType">
        <xs:sequence>
          <xs:choice>
            <xs:element name="Folder" type="t:FolderType"/>
            <xs:element name="CalendarFolder" type="t:CalendarFolderType"/>
            <xs:element name="ContactsFolder" type="t:ContactsFolderType"/>
            <xs:element name="SearchFolder" type="t:SearchFolderType"/>
            <xs:element name="TasksFolder" type="t:TasksFolderType"/>
          </xs:choice>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="ArrayOfFoldersType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="Folder" type="t:FolderType"/>
      <xs:element name="CalendarFolder" type="t:CalendarFolderType"/>
      <xs:element name="ContactsFolder" type="t:ContactsFolderType"/>
      <xs:element name="SearchFolder" type="t:SearchFolderType"/>
      <xs:element name="TasksFolder" type="t:TasksFolderType"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="ArrayOfUnknownEntriesType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="UnknownEntry" type="xs:string"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="BaseFolderIdType" abstract="true">
    <xs:annotation>
      <xs:documentation>Utility type which should never appear in user
documents</xs:documentation>
    </xs:annotation>
  </xs:complexType>
  <xs:complexType name="BaseFolderType" abstract="true">
    <xs:sequence>
      <xs:element name="FolderId" type="t:FolderIdType" minOccurs="0"/>
      <xs:element name="ParentFolderId" type="t:FolderIdType" minOccurs="0"/>
      <xs:element name="FolderClass" type="xs:string" minOccurs="0"/>
      <xs:element name="DisplayName" type="xs:string" minOccurs="0"/>
      <xs:element name="TotalCount" type="xs:int" minOccurs="0"/>
      <xs:element name="ChildFolderCount" type="xs:int" minOccurs="0"/>
```

```
      <xs:element name="ExtendedProperty" type="t:ExtendedPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="ManagedFolderInformation" type="t:ManagedFolderInformationType"
minOccurs="0"/>
      <xs:element name="EffectiveRights" type="t:EffectiveRightsType" minOccurs="0"/>
      <xs:element name="DistinguishedFolderId" type="t:DistinguishedFolderIdNameType"
minOccurs="0"/>
      <xs:element name="PolicyTag" type="t:RetentionTagType" minOccurs="0"/>
      <xs:element name="ArchiveTag" type="t:RetentionTagType" minOccurs="0"/>
      <xs:element name="ReplicaList" type="t:ArrayOfStringsType" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="BasePermissionType" abstract="true">
    <xs:sequence>
      <xs:element name="UserId" type="t:UserIdType"/>
      <xs:element name="CanCreateItems" type="xs:boolean" minOccurs="0"/>
      <xs:element name="CanCreateSubFolders" type="xs:boolean" minOccurs="0"/>
      <xs:element name="IsFolderOwner" type="xs:boolean" minOccurs="0"/>
      <xs:element name="IsFolderVisible" type="xs:boolean" minOccurs="0"/>
      <xs:element name="IsFolderContact" type="xs:boolean" minOccurs="0"/>
      <xs:element name="EditItems" type="t:PermissionActionType" minOccurs="0"/>
      <xs:element name="DeleteItems" type="t:PermissionActionType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ConsumerCalendarIdType">
    <xs:complexContent>
      <xs:extension base="t:BaseFolderIdType">
        <xs:attribute name="OwnerPuid" type="xs:long" use="required"/>
        <xs:attribute name="OwnerCid" type="xs:long" use="optional"/>
        <xs:attribute name="CalendarGuid" type="t:GuidType" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="DeleteFolderFieldType">
    <xs:complexContent>
      <xs:extension base="t:FolderChangeDescriptionType">
        <xs:sequence/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="FolderChangeDescriptionType">
    <xs:complexContent>
      <xs:extension base="t:ChangeDescriptionType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="FolderChangeType">
    <xs:sequence>
      <xs:choice>
        <xs:element name="FolderId" type="t:FolderIdType"/>
        <xs:element name="DistinguishedFolderId" type="t:DistinguishedFolderIdType"/>
      </xs:choice>
      <xs:element name="Updates" type="t:NonEmptyArrayOfFolderChangeDescriptionsType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="FolderClassType">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:complexType name="ManagedFolderInformationType">
    <xs:sequence>
      <xs:element name="CanDelete" type="xs:boolean" minOccurs="0"/>
      <xs:element name="CanRenameOrMove" type="xs:boolean" minOccurs="0"/>
      <xs:element name="MustDisplayComment" type="xs:boolean" minOccurs="0"/>
      <xs:element name="HasQuota" type="xs:boolean" minOccurs="0"/>
      <xs:element name="IsManagedFoldersRoot" type="xs:boolean" minOccurs="0"/>
      <xs:element name="ManagedFolderId" type="xs:string" minOccurs="0"/>
      <xs:element name="Comment" type="xs:string" minOccurs="0"/>
      <xs:element name="StorageQuota" type="xs:int" minOccurs="0"/>
      <xs:element name="FolderSize" type="xs:int" minOccurs="0"/>
      <xs:element name="HomePage" type="xs:string" minOccurs="0"/>
```

```
  </xs:sequence>
</xs:complexType>
<xs:complexType name="FolderType">
  <xs:complexContent>
    <xs:extension base="t:BaseFolderType">
      <xs:sequence>
        <xs:element name="PermissionSet" type="t:PermissionSetType" minOccurs="0"/>
        <xs:element name="UnreadCount" type="xs:int" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="NonEmptyArrayOfBaseFolderIdsType">
  <xs:choice maxOccurs="unbounded">
    <xs:element name="FolderId" type="t:FolderIdType"/>
    <xs:element name="DistinguishedFolderId" type="t:DistinguishedFolderIdType"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="NonEmptyArrayOfFolderChangeDescriptionsType">
  <xs:choice maxOccurs="unbounded">
    <xs:element name="AppendToFolderField" type="t:AppendToFolderFieldType"/>
    <xs:element name="SetFolderField" type="t:SetFolderFieldType"/>
    <xs:element name="DeleteFolderField" type="t:DeleteFolderFieldType"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="NonEmptyArrayOfFolderChangesType">
  <xs:sequence>
    <xs:element name="FolderChange" type="t:FolderChangeType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="NonEmptyArrayOfFolderNamesType">
  <xs:sequence>
    <xs:element name="FolderName" type="xs:string" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="NonEmptyArrayOfFoldersType">
  <xs:choice maxOccurs="unbounded">
    <xs:element name="Folder" type="t:FolderType"/>
    <xs:element name="CalendarFolder" type="t:CalendarFolderType"/>
    <xs:element name="ContactsFolder" type="t:ContactsFolderType"/>
    <xs:element name="SearchFolder" type="t:SearchFolderType"/>
    <xs:element name="TasksFolder" type="t:TasksFolderType"/>
  </xs:choice>
</xs:complexType>
<xs:simpleType name="PermissionActionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="Owned"/>
    <xs:enumeration value="All"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PermissionLevelType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="Owner"/>
    <xs:enumeration value="PublishingEditor"/>
    <xs:enumeration value="Editor"/>
    <xs:enumeration value="PublishingAuthor"/>
    <xs:enumeration value="Author"/>
    <xs:enumeration value="NoneditingAuthor"/>
    <xs:enumeration value="Reviewer"/>
    <xs:enumeration value="Contributor"/>
    <xs:enumeration value="Custom"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PermissionReadAccessType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="FullDetails"/>
```

```
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="PermissionSetType">
      <xs:annotation>
        <xs:documentation>The set of permissions on a folder</xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <xs:element name="Permissions" type="t:ArrayOfPermissionsType"/>
        <xs:element name="UnknownEntries" type="t:ArrayOfUnknownEntriesType" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="PermissionType">
      <xs:annotation>
        <xs:documentation>A permission on a folder</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
        <xs:extension base="t:BasePermissionType">
          <xs:sequence>
            <xs:element name="ReadItems" type="t:PermissionReadAccessType" minOccurs="0"/>
            <xs:element name="PermissionLevel" type="t:PermissionLevelType"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="SetFolderFieldType">
      <xs:complexContent>
        <xs:extension base="t:FolderChangeDescriptionType">
          <xs:choice>
            <xs:element name="Folder" type="t:FolderType"/>
            <xs:element name="CalendarFolder" type="t:CalendarFolderType"/>
            <xs:element name="ContactsFolder" type="t:ContactsFolderType"/>
            <xs:element name="SearchFolder" type="t:SearchFolderType"/>
            <xs:element name="TasksFolder" type="t:TasksFolderType"/>
          </xs:choice>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="TargetFolderIdType">
      <xs:choice>
        <xs:element name="FolderId" type="t:FolderIdType"/>
        <xs:element name="DistinguishedFolderId" type="t:DistinguishedFolderIdType"/>
        <xs:element name="AddressListId" type="t:AddressListIdType"/>
        <xs:element name="ConsumerCalendarId" type="t:ConsumerCalendarIdType"/>
      </xs:choice>
    </xs:complexType>
  </xs:schema>
```

8 Appendix C: Product Behavior

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

- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016

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

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

<1> Section 2.2.4.1: Exchange 2007 and Exchange 2010 do not support the **AddressListIdType** complex type.

<2> Section 2.2.4.5: Exchange 2007 and Exchange 2010 do not support the **DistinguishedFolderId** element in the **BaseFolderType** complex type.

<3> Section 2.2.4.5: Exchange 2007 and Exchange 2010 do not support the **PolicyTag** element.

<4> Section 2.2.4.5: Exchange 2007 and Exchange 2010 do not support the **ArchiveTag** element.

<5> Section 2.2.4.5: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **ReplicaList** element.

<6> Section 2.2.4.8: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **ConsumerCalendarIdType** complex type.

<7> Section 2.2.4.15: Exchange 2007 and Exchange 2010 will return an **ErrorInvalidPermissionSettings** ([MS-OXWSCDATA] section 2.2.5.24) response code if any field of **BasePermissionType** other than **UserId** field is set, and the **PermissionLevel** field is not set to "Custom".

<8> Section 2.2.4.16: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **ConsumerCalendarId** element.

<9> Section 2.2.5.3: In Microsoft Exchange Server 2013 Service Pack 1 (SP1) and Exchange 2016 the user can create items in the folder and read those items.

<10> Section 3.1.4.1: Exchange 2013 and Exchange 2016 do not support the **CopyFolder** operation if either the source folder or the destination folder is a **public folder**.

<11> Section 3.1.4.5: Exchange 2013 and Exchange 2016 do not support the **EmptyFolder** operation for use with public folders.

<12> Section 3.1.4.5: Exchange 2007 and the initial release version of Exchange 2010 do not include the **EmptyFolder** operation.

9 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

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

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

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2.4.5 t:BaseFolderType	Added a product behavior note for the ReplicaList child element.	N	New product behavior note added.
2.2.4.7 t:BasePermissionType	Updated the descriptions for the CanCreateItems, CanCreateSubFolders, EditItems, and DeleteItems child elements to indicate that IsFolderOwner also must be "true" for the client.	N	Content update.
2.2.4.8 t:ConsumerCalendarIdType	Added new complex type ConsumerCalendarIdType.	Y	New content added.
2.2.4.8 t:ConsumerCalendarIdType	Added a new section for this complex type.	Y	New content added.
2.2.4.16 t:TargetFolderIdType	Added details for the ConsumerCalendarId child element.	N	New content added.
2.2.4.16 t:TargetFolderIdType	Added a product behavior note for the ConsumerCalendarId child element.	N	New product behavior note added.
2.2.5.3 t:PermissionLevelType	Clarified the description of the "Contributor" value.	N	Content update.
2.2.5.3 t:PermissionLevelType	Added a product behavior note for the "Contributor" value.	N	New product behavior note added.
3.1.4.1 CopyFolder	Updated product behavior note for the latest product version.	N	Product behavior note updated.
3.1.4.5 EmptyFolder	Updated product behavior note for the latest product version.	N	Product behavior note updated.
6 Appendix A: Full WSDL	Updated WSDL to reflect the latest product version.	N	Content update.
7.1 Messages Schema	Updated schema to reflect the latest product version.	N	Content update.
7.2 Types Schema	Added details for the ReplicaList child element.	N	Content update.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
7.2 Types Schema	Added "ConsumerCalendarIdType" to code example.	N	Content update.
7.2 Types Schema	Updated schema to reflect the latest product version.	N	Content update.
8 Appendix C: Product Behavior	Updated the list of applicable products.	N	Content update.

10 Index