[MS-OXWSFOLD]:
Folders and Folder Permissions Web Service Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
07/15/2009	1.0	Major	Initial Availability.
11/04/2009	1.0.1	Editorial	Revised and edited the technical content.
02/10/2010	2.0.0	Major	Updated and revised the technical content.
05/05/2010	3.0.0	Major	Updated and revised the technical content.
08/04/2010	4.0	Major	Significantly changed the technical content.
11/03/2010	5.0	Major	Significantly changed the technical content.

Contents

Release: Tuesday, October 26, 2010

1   Introduction

This document specifies the Folders and Folder Permissions Web Service protocol, which is used by clients to manipulate and organize the folders that contain messages and subfolders This protocol specifies the way to retrieve and set permissions, the rights to access or to perform certain operations on the folder.

1.1   Glossary

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

Calendar folder
contact
folder
Folder object
Hypertext Transport Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
Junk E-mail folder
mailbox
message
permissions
property(1)
search folder
Sent Items folder
SOAP message
special folder
store
Web Services Description Language (WSDL)
WSDL message
WSDL port type
XML
XML namespace
XML schema

The following terms are specific to this document:

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

1.2   References

1.2.1   Normative References

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

[MS-OXDSCLI] Microsoft Corporation, "Autodiscover Publishing and Lookup Protocol Specification", April 2008.

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

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

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

[MS-OXWSGTZ] Microsoft Corporation, "Get Server Time Zone Web Service Protocol Specification", July 2009.

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

[MS-OXWSSRCH] Microsoft Corporation, "Mailbox Search Web Service Protocol Specification", July 2009.

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

[MS-OXWSXPROP] Microsoft Corporation, "Extended Properties Web Service Protocol Specification", July 2009.

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

[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", W3C Note, 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., Eds., et al., "Namespaces in XML 1.0 (Third Edition)", December 2009, http://www.w3.org/TR/REC-xml-names/

[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., and Malhotra, A., Eds., "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-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", April 2008.

1.3   Overview

The Folders and Folder Permissions Web Service protocol provides clients with the **folder** operations for retrieving folder **permission** lists that are stored on the server. The protocol specifies the elements and operations for creating and locating the **special folders** in a **mailbox**.

1.4   Relationship to Other Protocols

The Folders and Folder Permissions Web Service protocol uses SOAP over **HTTP** and SOAP over **HTTPS**, as shown in the following figures.
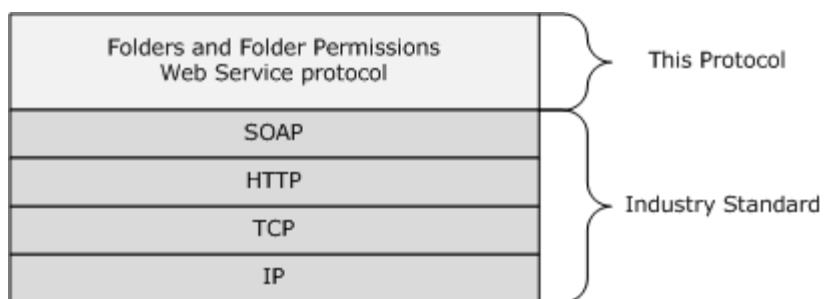
Figure 1: Folders and Folder Permissions Web Service protocol HTTP stack
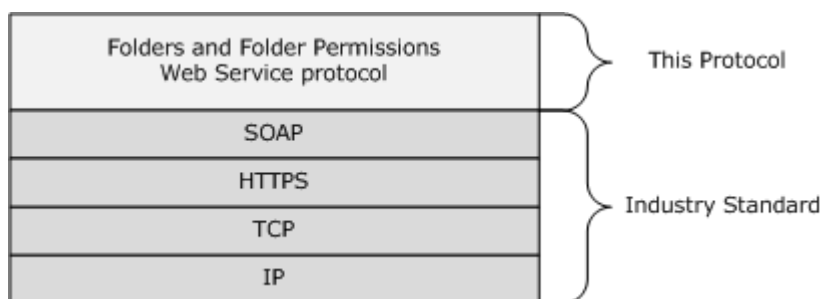
Figure 2: Folders and Folder Permissions Web Service protocol HTTPS stack

1.5   Prerequisites/Preconditions

None.

1.6   Applicability Statement

The specification is applicable to environments that copy, create, delete, get, move, or update folders.

1.7   Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

▪ **Supported Transports:** This protocol uses SOAP 1.1.

▪ **Protocol Versions:** This protocol has only one **WSDL port type** version.

▪ **Security and Authentication Methods:** This protocol relies on the Web server that hosts the application to perform authentication. The server will not allow unauthenticated clients to access the Voice Mail Settings Web Service protocol operations.

▪ **Capability Negotiation:** This protocol does not perform explicit capability negotiation. This protocol relies on the Autodiscover Publishing and Lookup protocol [MS-OXDSCLI] to identify protocol versions and service **endpoint** that are applicable to a client.

1.8   Vendor-Extensible Fields

None.

1.9   Standards Assignments

None.

2 Messages

2.1 Transport

Messages are transported by using SOAP version 1.1. For details, see [SOAP1.1].

2.2 Common Message Syntax

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

2.2.1 Namespaces

This specification defines and references various **XML namespaces** by using the mechanisms specified in [XMLNS]. Although this specification associates a specific XML namespace prefix 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	[XMLSCHEMA2]
soap	http://schemas.xmlsoap.org/WSDL/soap/	[WSDL]
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** 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 defined by this specification. XML schema complex types that are specific to a particular operation are described with the operation.

ComplexType	Description
ArrayOfFoldersType	Specifies an array of folders.
BaseMoveCopyFolderType	The base type for the **CopyFolderType** (section 3.1.4.1.3.2) and **MoveFolderType** (section 3.1.4.7.3.2) complex types.

ComplexType	Description
BasePermissionType	The base type that defines base permissions for items and 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.
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:ArrayOfFoldersType Complex Type

The **ArrayOfFoldersType** complex type specifies an array of folders.

```
<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>
```

Child Elements

Element	Type	Description
Folder	**t:FolderType**	Represents a regular folder in the server database.
CalendarFolder	**t:CalendarFolderType**	Represents a **Calendar folder** in a mailbox. The **CalendarFolderType** complex type is specified in [MS-OXWSMTGS] section 2.2.3.3.
ContactsFolder	**t:ContactsFolderType**	Represents a Contacts folder in a mailbox. The **ContactsFolderType** complex type is specified in [MS-OXWSCONT] section 3.1.4.1.2.2.
SearchFolder	**t:SearchFolderType**	Represents a **search folder** that is contained in a mailbox. The **SearchFolderType** complex type is specified in [MS-OXWSSRCH] section 2.2.3.26.
TasksFolder	**t:TasksFolderType**	Represents a Tasks folder that is contained in a mailbox. The **TasksFolderType** complex type is specified in [MS-OXWSTASK] section 2.2.3.2.

2.2.4.2   m:BaseMoveCopyFolderType Complex Type

The **BaseMoveCopyFolderType** complex type is the base type for the **CopyFolderType** (section 3.1.4.1.3.2) and **MoveFolderType** (section 3.1.4.7.3.2) complex types. The **BaseMoveCopyFolderType** complex type extends the **BaseRequestType** complex type ([MS-OXWSCDATA] section 2.2.3.14).

```
<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>
```

Child Elements

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

2.2.4.3   t:BasePermissionType Complex Type

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

```xml
<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>
```

Child Elements

Release: Tuesday, October 26, 2010

Element	Type	Description
UserId	**t:UserIdType**	Specifies a user identifier. The **UserIdType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.57.
CanCreateItems	xs:boolean	Specifies whether a client can create items in a folder. The **boolean** data type is specified in [XMLSCHEMA2].
CanCreateSubFolders	xs:boolean	Specifies whether the client can create subfolders.
IsFolderOwner	xs:boolean	Specifies whether the user is the owner of a folder.
IsFolderVisible	xs:boolean	Specifies whether a user can view a folder.
IsFolderContact	xs:boolean	Specifies whether a user is a contact for a folder.
EditItems	**t:PermissionActionType**	Specifies whether a client can edit items.
DeleteItems	**t:PermissionActionType**	Specifies whether the client can delete items.

The **PermissionType** complex type ([MS-OXWSMTGS] section 2.2.4.13) and the **CalendarPermissionType** complex type ([MS-OXWSMTGS] section 2.2.3.6) extend the **BasePermissionType** complex type.

2.2.4.4   t:ArrayOfUnknownEntriesType Complex Type

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>
```

Child Elements

Element	Type	Description
UnknownEntry	xs:string	Represents an array of unknown permission entries that cannot be resolved against the directory service. The **string** data type is specified in [XMLSCHEMA2].

2.2.4.5 t:BaseFolderIdType Complex Type

The **BaseFolderIdType** complex type specifies the base type for derived types that represent a folder identifier. The **BaseFolderIdType** complex type MUST NOT be sent in a **SOAP message** because it is an abstract type.

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

The **DistinguishedFolderIdType** complex type ([MS-OXWSCDATA] section 2.2.3.21) and the **FolderIdType** complex type ([MS-OXWSCDATA] section 2.2.3.29) extend the **BaseFolderIdType** complex type.

2.2.4.6 t:BaseFolderType Complex Type

The **BaseFolderType** complex type specifies the base type for folders. The **BaseFolderType** complex type MUST NOT be sent in a SOAP message because it is an abstract type.

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

Release: Tuesday, October 26, 2010

```
      <xs:element name="EffectiveRights"
        type="t:EffectiveRightsType"
        minOccurs="0"
      />
    </xs:sequence>
  </xs:complexType>
```

Child Elements

Element	Type	Description
FolderId	**FolderIdType**	Specifies the folder identifier and change key. The **FolderIdType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.29.
ParentFolderId	**FolderIdType**	Specifies the parent folder identifier.
FolderClass	xs:string	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. The **string** data type is specified in [XMLSCHEMA2].
DisplayName	xs:string	Specifies the display name of the folder.
TotalCount	xs:int	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. The **int** data type is specified in [XMLSCHEMA2].
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.
ExtendedProperty	**t:ExtendedPropertyType**	Specifies the set of extended properties on a folder. The **ExtendedPropertyType** complex type is specified in [MS-OXWSXPROP] section 2.2.4.2.
ManagedFolderInformation	**t:ManagedFolderInformationType**	Specifies metadata for a managed folder. This property MUST be read-

Element	Type	Description
		only for a client and is only returned in a response.
EffectiveRights	**t:EffectiveRightsType**	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. The **EffectiveRightsType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.23.

The **FolderType** complex type (section 2.2.4.10), **CalendarFolderType** complex type ([MS-OXWSMTGS] section 2.2.3.3), and **ContactsFolderType** complex type ([MS-OXWSCONT] section 3.1.4.1.2.2) extend the **BaseFolderType** complex type.

2.2.4.7   t:FolderChangeDescriptionType Complex Type

The **FolderChangeDescriptionType** complex type specifies a change to a single folder property.

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

The **SetFolderFieldType** complex type (section 3.1.4.8.3.7), the **DeleteFolderFieldType** complex type (section 3.1.4.8.3.4), and the **AppendToFolderFieldType** complex type (section 3.1.4.8.3.3) extend the **FolderChangeDescriptionType** complex type.

2.2.4.8   t:FolderChangeType Complex Type

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>
```

Child Elements

Element	Type	Description
FolderId	**t:FolderIdType**	Specifies an identifier for a folder. The **FolderIdType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.29.
DistinguishedFolderId d	**t:DistinguishedFolderIdType**	Specifies an identifier for a distinguished folder. The **DistinguishedFolderIdType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.21.
Updates	**t:NonEmptyArrayOfFolderChangeDescriptionsType**	Specifies a collection of changes to a folder.

2.2.4.9 m:FolderInfoResponseMessageType Complex Type

The **FolderInfoResponseMessageType** complex type represents the response message for the **CreateFolder** operation (section 3.1.4.2), the **GetFolder** operation (section 3.1.4.6), the **UpdateFolder** operation (section 3.1.4.8), the **MoveFolder** operation (section 3.1.4.7), the **CopyFolder** operation (section 3.1.4.1), and the **CreateManagedFolder** operation (section 3.1.4.3).

```
<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>
```

Child Elements

Element	Type	Description
Folders	**t:ArrayOfFoldersType**	Represents the folders that are returned with the response message.

2.2.4.10   t:FolderType Complex Type

The **FolderType** complex type represents a regular folder in the server database.

```
<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>
```

Child Elements

Element	Type	Description
PermissionSet	**t:PermissionSetType**	Specifies all permissions that are configured for a folder.
UnreadCount	xs:int	Specifies the number of unread items in a folder. This element MUST only exist in responses and MUST equal the sum of all MessageType and PostItemType item types that have the IsRead property set to **false**. The **int** data type is specified in [XMLSCHEMA2].

The **SearchFolderType** ([MS-OXWSSRCH] section 2.2.3.26) and the **TasksFolderType** ([MS-OXWSTASK] section 2.2.3.2) extend the **FolderType** complex type.

2.2.4.11   t:ManagedFolderInformationType Complex Type

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>
```

Child Elements

Element	Type	Description
CanDelete	xs:boolean	A value that indicates whether a managed folder can be deleted by a client. The **boolean** data type is specified in [XMLSCHEMA2].
CanRenameOrMove	xs:boolean	A value that indicates whether a managed folder can be renamed or moved by a client.
MustDisplayComment	xs:boolean	A value that indicates whether the managed folder comment is to be displayed.
HasQuota	xs:boolean	A value that indicates whether a managed folder has a quota.
IsManagedFoldersRoot	xs:boolean	A value that indicates whether the managed folder is the root managed folder.
ManagedFolderId	xs:string	The unique identifier of a managed folder. The **string** data type is specified in [XMLSCHEMA2].
Comment	xs:string	A comment that is associated with a managed folder.
StorageQuota	xs:int	The storage quota for a managed folder. The **int** data type is

Element	Type	Description
		specified in [XMLSCHEMA2].
FolderSize	xs:int	A value that describes the total size of all the contents of a managed folder.
HomePage	xs:string	The URL that is the default home page for the managed folder.

2.2.4.12   t:PermissionSetType Complex Type

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>
```

Child Elements

Element	Type	Description
Permissions	**t:ArrayOfPermissionsType**	Specifies a collection of permissions for a folder. The **ArrayOfPermissionsType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.6.
UnknownEntries	**t:ArrayOfUnknownEntriesType**	Represents an array of unknown permission entries that cannot be resolved against the directory service.

2.2.4.13   t:PermissionType Complex Type

The **PermissionType** complex type specifies a permission on a folder.

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

Child Elements

Element	Type	Description
ReadItems	**t:PermissionReadAccessType**	Specifies whether a user has permission to read items in a folder.
PermissionLevel	**t:PermissionLevelType**	Specifies the combination of permissions that a user has on a folder.

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

2.2.4.14   t:TargetFolderIdType Complex Type

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

Child Elements

Element	Type	Description
FolderId	**t:FolderIdType**	Specifies a folder identifier. The **FolderIdType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.29.
DistinguishedFolderId	**t:DistinguishedFolderIdType**	Specifies a distinguished folder identifier. The **DistinguishedFolderIdType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.21.

This type is a container for either a **FolderIdType** complex type or **DistinguishedFolderIdType** complex type object.

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	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 Simple Type

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

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

2.2.5.2 t:PermissionActionType Simple Type

The **PermissionActionType** simple type enumeration 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>
```

Enumeration

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

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

The **PermissionLevelType** simple type enumeration 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>
```

Enumeration

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

Value	Description
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.
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. The contents of the folder do not appear.
Custom	The user has custom access permissions on the folder.

2.2.5.4   t:PermissionReadAccessType Simple Type

The **PermissionReadAccessType** simple type enumeration 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>
```

Enumeration

The following values 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

This protocol defines a single WSDL port type that contains a set of operations as listed in the following table.

Operation	Description
CopyFolder	Copies identified folders and returns the folder identifier and change key of the copied folders.
CreateFolder	Creates folders, Calendar folders, Contacts folders, Tasks folders, and search folders.
CreateManagedFolder	Creates a managed folder in the server **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 store.
MoveFolder	Moves folders from a specified folder and puts them in another folder.
UpdateFolder	Modifies properties of an existing item in the server store.

3.1.1   Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that specified in this document.

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	Description
CopyFolder	Copies identified folders and returns the folder identifier and change key of the copied folders.

Operation	Description
CreateFolder	Creates folders, Calendar folders, Contacts folders, Tasks folders, and search folders.
CreateManagedFolder	Creates a managed folder in the server 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 store.
MoveFolder	Moves folders from a specified folder and puts them in another folder.
UpdateFolder	Modifies properties of an existing item in the server store.

3.1.4.1 CopyFolder Operation

The **CopyFolder** operation copies identified folders and returns the folder identifier and change key of the copied folders.

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

Request

Message Format	Description
tns:CopyFolderSoapIn	Specifies the SOAP message that copies the folders.

Response

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

3.1.4.1.1 Messages

The following WSDL message definitions are specific to this operation.

Messages	Description
CopyFolderSoapIn	Specifies the request that copies a folder.
CopyFolderSoapOut	Specifies the response from a request to copy a folder.

3.1.4.1.1.1 tns:CopyFolderSoapIn Message

The **CopyFolderSoapIn** WSDL message specifies the **CopyFolder** operation request.

```
<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 four parts of the **CopyFolderSoapIn** message are described in the following table.

Part Name	Element/Type	Description
request	<tns:CopyFolder>	Specifies the request message.
Impersonation	<t:ExchangeImpersonation>	Specifies the identifier of the account to impersonate. The <ExchangeImpersonation> element is specified in [MS-OXWSCDATA] section 2.2.4.2.
MailboxCulture	<t:MailboxCulture>	Specifies the culture to use for accessing the server. The cultures are defined by [RFC3066]. The <MailboxCulture> element is specified in [MS-OXWSCDATA] section 2.2.4.5.
RequestVersion	<t:RequestServerVersion>	Specifies the schema version for the **CopyFolder** operation request. The <RequestServerVersion> element is specified in [MS-OXWSCDATA] section 2.2.4.7.

3.1.4.1.1.2   tns:CopyFolderSoapOut Message

The **CopyFolderSoapOut** WSDL message specifies the **CopyFolder** operation response.

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

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

Part name	Element/type	Description
CopyFolderResult	<tns:CopyFolderResponse>	Specifies the response message.
ServerVersion	<t:ServerVersionInfo>	Specifies the server version for the response. The <ServerVersionInfo> element is specified in [MS-OXWSCDATA] section 2.2.4.8.

3.1.4.1.2   Elements

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

Element	Description
<CopyFolder>	Defines a request to copy folders in a mailbox in the server store.

Element	Description
<CopyFolderResponse>	Defines a response to a **CopyFolder** operation request.

3.1.4.1.2.1 CopyFolder Element

The <CopyFolder> element defines a request to copy folders in a mailbox in the server store.

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

3.1.4.1.2.2 CopyFolderResponse Element

The <CopyFolderResponse> element defines a response to a **CopyFolder** operation (section 3.1.4.1).

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

3.1.4.1.3 Complex Types

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

Type	Description
CopyFolderResponseType	Represents a response to a **CopyFolder** operation.
CopyFolderType	Represents an operation to copy folders in a server database.

3.1.4.1.3.1 m:CopyFolderResponseType Complex Type

The **CopyFolderResponseType** complex type represents a response to a **CopyFolder** operation. The **CopyFolderResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.3.15).

```
<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 represents an operation to copy folders in a server database. The **CopyFolderType** complex type extends the **BaseMoveCopyFolderType** complex type as specified in section 2.2.4.2.

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

3.1.4.2   CreateFolder Operation

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

Before any data can be read from or written to a folder, an implementation needs to ensure that the folder exists and open it, or create 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.

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

Request

Message Format	Description
tns:CreateFolderSoapIn	Specifies the SOAP message that creates the folders.

Response

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

3.1.4.2.1   Messages

The following WSDL message definitions are specific to this operation.

Messages	Description
CreateFolderSoapIn	Specifies the request that creates a folder.
CreateFolderSoapOut	Specifies the response from a request to create a folder.

3.1.4.2.1.1   tns:CreateFolderSoapIn Message

The **CreateFolderSoapIn** WSDL message specifies the **CreateFolder** operation request.

```
<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 five parts of the **CreateFolderSoapIn** message are described in the following table.

Part Name	Element/Type	Description
request	<tns:CreateFolder>	Specifies the request message.
Impersonation	<t:ExchangeImpersonation>	Specifies the identifier of the account to impersonate. The <ExchangeImpersonation> element is specified in [MS-OXWSCDATA] section 2.2.4.2.
MailboxCulture	<t:MailboxCulture>	Specifies the culture to use for accessing the server. The cultures are defined by [RFC3066]. The <MailboxCulture> element is specified in [MS-OXWSCDATA] section 2.2.4.5.
RequestVersion	<t:RequestServerVersion>	Specifies the schema version for the **CreateFolder** operation request. The <RequestServerVersion> element is specified in [MS-OXWSCDATA] section 2.2.4.7.
TimeZoneContext	<t:TimeZoneContext>	Specifies a time zone definition and enables the association of SOAP attributes with the definition. The <TimeZoneContext> element is specified in [MS-OXWSGTZ] section 2.2.4.4.

3.1.4.2.1.2   tns:CreateFolderSoapOut Message

The **CreateFolderSoapOut** WSDL message specifies the **CreateFolder** operation response.

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

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

Part name	Element/type	Description
CreateFolderResult	<tns:CreateFolderResponse>	Specifies the response message.
ServerVersion	<t:ServerVersionInfo>	Specifies the server version for the response. The <ServerVersionInfo> element is specified in [MS-

Part name	Element/type	Description
		OXWSCDATA] section 2.2.4.8.

3.1.4.2.2 Elements

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

Element	Description
<CreateFolder>	Defines a request to create folders in a mailbox in the server store.
<CreateFolderResponse>	Defines a response to a **CreateFolder** operation request.

3.1.4.2.2.1 CreateFolder Element

The <CreateFolder> element defines a request to create a folder in a mailbox in the server store.

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

3.1.4.2.2.2 CreateFolderResponse Element

The <CreateFolderResponse> element defines a response to a **CreateFolder** operation request.

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

3.1.4.2.3 Complex Types

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

Type	Description
CreateFolderResponseType	Represents a response to a **CreateFolder** operation.
CreateFolderType	Represents an operation to create folders in a server database.
NonEmptyArrayOfFoldersType	An array of folders that has at least one member.

3.1.4.2.3.1 t:CreateFolderResponseType Complex Type

The **CreateFolderResponseType** complex type represents a response to a **CreateFolder** operation. The **CreateFolderResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.3.15).

```
<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 represents a request to create a folder.

```
<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>
```

Child Elements

Element	Type	Description
ParentFolderId	**t:TargetFolderIdType**	The identifier of the folder that will contain the newly created folder. The **TargetFolderIdType** complex type is specified in section 2.2.4.14.
Folders	**t:NonEmptyArrayOfFoldersType**	Represents a target folder for operations that create, save, copy, move, or synchronize items or folder.

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
    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>
```

Child Elements

Element	Type	Description
Folder	**t:FolderType**	Represents a regular folder in the server database. The **FolderType** complex type is specified in section 2.2.4.10.
CalendarFolder	**t:CalendarFolderType**	Represents a folder that primarily contains calendar items. The **CalendarFolderType** complex type is specified in [MS-OXWSMTGS] section 2.2.3.3.
ContactsFolder	**t:ContactsFolderType**	Represents a Contacts folder in a mailbox. The **ContactsFolderType** complex type is specified in [MS-OXWSCONT] section 3.1.4.1.2.2.
SearchFolder	**t:SearchFolderType**	Represents a search folder that is contained in a mailbox. The **SearchFolderType** complex type is specified in [MS-OXWSSRCH] section 2.2.3.26.
TasksFolder	**t:TasksFolderType**	Represents a Tasks folder that is contained in a mailbox. The **TasksFolderType** complex type is specified in [MS-OXWSTASK] section 2.2.3.2.

3.1.4.3 CreateManagedFolder Operation

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

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

Request

Message Format	Description
tns:CreateManagedFolderSoapIn	Specifies the SOAP message that creates the managed folder.

Response

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

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.

3.1.4.3.1 Messages

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

Message	Description
CreateManagedFolderSoapIn	Specifies the request that creates a managed folder.
CreateManagedFolderSoapOut	Specifies the response from a request to create a managed folder.

3.1.4.3.1.1 tns:CreateManagedFolderSoapIn Message

The **CreateManagedFolderSoapIn** WSDL message specifies the **CreateManagedFolder** operation request.

```
<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 four parts of the **CreateManagedFolderSoapIn** message are described in the following table.

Part Name	Element/Type	Description
request	<tns:CreateManagedFolder>	Specifies the request message.
Impersonation	<t:ExchangeImpersonation>	Specifies the identifier of the account to impersonate. The <ExchangeImpersonation> element is specified in [MS-OXWSCDATA] section 2.2.4.2.
MailboxCulture	<t:MailboxCulture>	Specifies the culture to use for accessing the server. The cultures are defined by [RFC3066]. The <MailboxCulture> element is specified in [MS-OXWSCDATA] section 2.2.4.5.
RequestVersion	<t:RequestServerVersion>	Specifies the schema version for the **CreateManagedFolder** request. The <RequestServerVersion> element is specified in [MS-OXWSCDATA] section 2.2.4.7.

3.1.4.3.1.2 tns:CreateManagedFolderSoapOut Message

The **CreateManagedFolderSoapOut** WSDL message specifies the **CreateManagedFolder** operation response.

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

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

Part name	Element/type	Description
CreateManagedFolderResult	<tns:CreateManagedFolderResponse>	Specifies the response message.
ServerVersion	<t:ServerVersionInfo>	Specifies the server version for the response. The <ServerVersionInfo> element is specified in [MS-OXWSCDATA] section 2.2.4.8.

3.1.4.3.2 Elements

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

Element	Description
<CreateManagedFolder>	Defines a request to add a managed folder to a mailbox in the server store.
<CreateManagedFolderResponse>	Defines a response to a **CreateManagedFolder** operation request.

3.1.4.3.2.1 CreateManagedFolder Element

The <CreateManagedFolder> element defines a request to create a managed folder in a mailbox in the server store.

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

3.1.4.3.2.2 CreateManagedFolderResponse Element

The <CreateManagedFolderResponse> element defines a response to a **CreateManagedFolder** operation request.

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

3.1.4.3.3 Complex Types

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

Type	Description
CreateManagedFolderRequestType	Represents an operation to create a managed folder in a server database.
CreateManagedFolderResponseType	Represents a response to a **CreateManagedFolder** operation.
NonEmptyArrayOfFolderNamesType	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 represents an operation to create a managed folder in a server database.

```
<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>
```

Child Elements

Element	Type	Description
FolderNames	**t:NonEmptyArrayOfFolderNamesType**	An array of managed folders to add to a mailbox. This array has at least one member.
Mailbox	**t:EmailAddressType**	The e-mail address of the mailbox in which the managed folders are added. The **EmailAddressType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.25.

3.1.4.3.3.2 m:CreateManagedFolderResponseType Complex Type

The **CreateManagedFolderResponseType** complex type represents a response to a **CreateManagedFolder** operation. The **CreateManagedFolderResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.3.15).

```
<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>
```

Child Elements

Element	Type	Description
FolderName	xs:string	An array of managed folders. The **string** type is specified in [XMLSCHEMA2].

3.1.4.4   DeleteFolder Operation

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.

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

Request

Message Format	Description
tns:DeleteFolderSoapIn	Specifies the SOAP message that deletes a folder.

Response

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

3.1.4.4.1 Messages

The following WSDL message definitions are specific to this operation.

Message	Description
DeleteFolderSoapIn	Specifies the request that deletes a folder.
DeleteFolderSoapOut	Specifies the response from a request to delete a folder.

3.1.4.4.1.1 tns:DeleteFolderSoapIn Message

The **DeleteFolderSoapIn** WSDL message specifies the **DeleteFolder** operation request.

```
<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 four parts of the **DeleteFolderSoapIn** message are described in the following table.

Part Name	Element/Type	Description
request	<tns:DeleteFolder>	Specifies the request message.
Impersonation	<t:ExchangeImpersonation>	Specifies the identifier of the account to impersonate. The <ExchangeImpersonation> element is specified in [MS-OXWSCDATA] section 2.2.4.2.
MailboxCulture	<t:MailboxCulture>	Specifies the culture to use for accessing the server. The cultures are defined by [RFC3066]. The <MailboxCulture> element is specified in [MS-OXWSCDATA] section 2.2.4.5.
RequestVersion	<t:RequestServerVersion>	Specifies the schema version for the **DeleteFolder** operation request. The <RequestServerVersion> element is specified in [MS-OXWSCDATA] section 2.2.4.7.

3.1.4.4.1.2 tns:DeleteFolderSoapOut Message

The **DeleteFolderSoapOut** WSDL message specifies the **DeleteFolder** operation response.

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

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

Part name	Element/type	Description
DeleteFolderResult	<tns:DeleteFolderResponse>	Specifies the response message.
ServerVersion	<t:ServerVersionInfo>	Specifies the server version for the response. The <ServerVersionInfo> element is specified in [MS-OXWSCDATA] section 2.2.4.8.

3.1.4.4.2 Elements

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

Element	Description
<DeleteFolder>	Defines a request to delete a folder from a mailbox in the server store.
<DeleteFolderResponse>	Defines a response to a **DeleteFolder** operation request.

3.1.4.4.2.1 DeleteFolder Element

The <DeleteFolder> element defines a request to delete a folder from a mailbox in the server store.

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

3.1.4.4.2.2 DeleteFolderResponse Element

The <DeleteFolderResponse> element represents a response to a **DeleteFolder** operation.

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

3.1.4.4.3 Complex Types

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

Type	Description
DeleteFolderType	Represents an operation to delete a folder in a server database.
DeleteFolderResponseType	Represents a response to a **DeleteFolder** operation.

3.1.4.4.3.1 m:DeleteFolderResponseType Complex Type

The **DeleteFolderResponseType** complex type represents a response to a **DeleteFolder** operation. The **DeleteFolderResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.3.15).

```
<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 represents a request to delete folders from a mailbox.

```
<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>
```

Child Elements

Element	Type	Description
FolderIds	**t:NonEmptyArrayOfBaseFolderIdsType**	An array of folders to be deleted from a mailbox. This array has at least one member. The **NonEmptyArrayOfBaseFolderIdsType** complex type is specified in section 3.1.4.6.3.3.

Attributes

Name	Type	Description
DeleteType	**t:DisposalType**	Describes how a folder is to be deleted. The **DisposalType** simple type is specified in [MS-OXWSCDATA] section 2.2.2.8.

3.1.4.5   EmptyFolder Operation

The **EmptyFolder** operation empties identified folders 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.<1>

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

Request

Message Format	Description
tns:EmptyFolderSoapIn	Specifies the SOAP message that empties the folders.

Response

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

3.1.4.5.1 Messages

The following WSDL message definitions are specific to this operation.

Message	Description
EmptyFolderSoapIn	Specifies the request that empties a folder.
EmptyFolderSoapOut	Specifies the response from a request to empty a folder.

3.1.4.5.1.1 tns:EmptyFolderSoapIn Message

The **EmptyFolderSoapIn** WSDL message specifies the **EmptyFolder** operation request.

```
<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 four parts of the **EmptyFolderSoapIn** message are described in the following table.

Part Name	Element/Type	Description
request	<tns:EmptyFolder>	Specifies the request message.
Impersonation	<t:ExchangeImpersonation>	Specifies the identifier of the account to impersonate. The <ExchangeImpersonation> element is specified in [MS-OXWSCDATA] section 2.2.4.2.
MailboxCulture	<t:MailboxCulture>	Specifies the culture to use for accessing the server. The cultures are defined by [RFC3066]. The <MailboxCulture> element is specified in [MS-OXWSCDATA] section 2.2.4.5.

Part Name	Element/Type	Description
RequestVersion	<t:RequestServerVersion>	Specifies the schema version for the **EmptyFolder** operation request. The <RequestServerVersion> element is specified in [MS-OXWSCDATA] section 2.2.4.7.

3.1.4.5.1.2 tns:EmptyFolderSoapOut Message

The **EmptyFolderSoapOut** WSDL message specifies the **EmptyFolder** operation response.

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

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

Part Name	Element/Type	Description
EmptyFolderResult	<tns:EmptyFolderResponse>	Specifies the response message.
ServerVersion	<t:ServerVersionInfo>	Specifies the server version for the response. The <ServerVersionInfo> element is specified in [MS-OXWSCDATA] section 2.2.4.8.

3.1.4.5.2 Elements

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

Element	Description
<EmptyFolder>	Defines a request to empty folders in a mailbox in the server store.
<EmptyFolderResponse>	Defines a response to an **EmptyFolder** operation request.

3.1.4.5.2.1 EmptyFolder Element

The <EmptyFolder> element defines a request to empty folders in a mailbox in the server store.

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

3.1.4.5.2.2 EmptyFolderResponse Element

The <EmptyFolderResponse> element defines a response to an **EmptyFolder** operation.

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

3.1.4.5.3 ComplexTypes

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

Type	Description
EmptyFolderType	Represents an operation to empty folders in a server database.
EmptyFolderResponseType	Represents a response to an **EmptyFolder** operation.

3.1.4.5.3.1 m:EmptyFolderType Complex Type

The **EmptyFolderType** complex type represents a request to empty a folder in a mailbox.

```
<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>
```

Child Elements

Element	Type	Description
FolderIds	t:NonEmptyArrayOfBaseFolderIdsType	Specifies the collection of items and folders to delete. The **NonEmptyArrayOfBaseFolderIdsType** complex type is specified in section 3.1.4.6.3.3.

Attributes

Name	Type	Description
DeleteType	**t:DisposalType**	Specifies an enumeration value that describes how an item is deleted. The **DisposalType** simple type is specified in [MS-

Name	Type	Description
		OXWSCDATA] section 2.2.2.8.
DeleteSubFolders	xs:boolean	A value that indicates whether the subfolders to also be deleted. True if the subfolders are to be deleted; otherwise, false. The **boolean** data type is specified in [XMLSCHEMA2].

3.1.4.5.3.2   m:EmptyFolderResponseType Complex Type

The **EmptyFolderResponseType** complex type represents a response to an **EmptyFolder** operation. The **EmptyFolderResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.3.15).

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

3.1.4.6   GetFolder Operation

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

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

Request

Message Format	Description
tns:GetFolderSoapIn	Specifies the SOAP message that gets folders from the server store.

Response

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

3.1.4.6.1   Messages

The following WSDL message definitions are specific to this operation.

Message	Description
GetFolderSoapIn	Specifies the request that gets a folder.

Message	Description
GetFolderSoapOut	Specifies the response from a request to get a folder.

3.1.4.6.1.1 tns:GetFolderSoapIn Message

The **GetFolderSoapIn** WSDL message specifies the **GetFolder** operation request.

```
<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:message>
```

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

Part Name	Element/Type	Description
request	<tns:GetFolder>	Specifies the request message.
Impersonation	<t:ExchangeImpersonation>	Specifies the identifier of the account to impersonate. The <ExchangeImpersonation> element is specified in [MS-OXWSCDATA] section 2.2.4.2.
MailboxCulture	<t:MailboxCulture>	Specifies the culture to use for accessing the server. The cultures are defined by [RFC3066]. The <MailboxCulture> element is specified in [MS-OXWSCDATA] section 2.2.4.5.
RequestVersion	<t:RequestServerVersion>	Specifies the schema version for the **GetFolder** operation request. The <RequestServerVersion> element is specified in [MS-OXWSCDATA] section 2.2.4.7.
TimeZoneContext	<t:TimeZoneContext>	Specifies a time zone definition and enables the association of SOAP attributes with the definition. The <TimeZoneContext> element is specified in [MS-OXWSGTZ] section 2.2.4.4.

3.1.4.6.1.2 tns:GetFolderSoapOut Message

The **GetFolderSoapOut** WSDL message specifies the **GetFolder** operation response.

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

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

Part name	Element/type	Description
GetFolderResult	<tns:GetFolderResponse>	Specifies the response message.
ServerVersion	<t:ServerVersionInfo>	Specifies the server version for the response. The <ServerVersionInfo> element is specified in [MS-OXWSCDATA] section 2.2.4.8.

3.1.4.6.2 Elements

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

Element	Description
<GetFolder>	Defines a request to get a folder from a mailbox in the server store.
<GetFolderResponse>	Defines a response to a **GetFolder** operation request.

3.1.4.6.2.1 GetFolder Element

The <GetFolder> element defines a request to get a folder from a mailbox in the server store.

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

3.1.4.6.2.2 GetFolderResponse Element

The <GetFolderResponse> element represents a response to a **GetFolder** operation request.

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

3.1.4.6.3 Complex Types

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

Type	Description
GetFolderResponseType	Represents a response to a **GetFolder** operation.
GetFolderType	Represents an operation to get a folder in a server database.
NonEmptyArrayOfBaseFolderIdsType	Specifies an array of one or more folders.

3.1.4.6.3.1 m:GetFolderResponseType Complex Type

The **GetFolderResponseType** complex type represents a response to a **GetFolder** operation. The **GetFolderResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.3.15).

```
<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 represents an operation to get a folder in a server database.

```
<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>
```

Child Elements

Element	Type	Description
FolderShape	**t:FolderResponseShapeType**	Specifies the properties to include in the response. The **FolderResponseShapeType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.30.
FolderIds	**t:NonEmptyArrayOfBaseFolderIdsType**	An array of one or more folder identifiers. The **NonEmptyArrayOfBaseFolderIdsType** complex type is specified in section 3.1.4.6.3.3.

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>
  <xs:choice
    maxOccurs="unbounded"
  >
```

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

Child Elements

Element	Type	Description
FolderId	**t:FolderIdType**	Identifies a folder. The **FolderIdType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.29.
DistinguishedFolderId	**t:DistinguishedFolderIdType**	Specifies a distinguished folder identifier. The **DistinguishedFolderIdType** complex type is specified in [MS-OXWSCDATA] section 2.2.3.21.

3.1.4.7   MoveFolder Operation

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.

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

Request

Message Format	Description
tns:MoveFolderSoapIn	Specifies the SOAP message that moves folders from a specified folder and puts them in another folder.

Response

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

3.1.4.7.1   Messages

The following WSDL message definitions are specific to this operation.

Message	Description
MoveFolderSoapIn	Specifies the request that moves a folder.

Message	Description
MoveFolderSoapOut	Specifies the response to a request to move a folder.

3.1.4.7.1.1 tns:MoveFolderSoapIn Message

The **MoveFolderSoapIn** WSDL message specifies the **MoveFolder** operation request.

```
<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 four parts of the **MoveFolderSoapIn** message are described in the following table.

Part Name	Element/Type	Description
request	<tns:MoveFolder>	Specifies the request message.
Impersonation	<t:ExchangeImpersonation>	Specifies the identifier of the account to impersonate. The <ExchangeImpersonation> element is specified in [MS-OXWSCDATA] section 2.2.4.2.
MailboxCulture	<t:MailboxCulture>	This part specifies the culture to use for accessing the server. The cultures are defined by [RFC3066]. The <MailboxCulture> element is specified in [MS-OXWSCDATA] section 2.2.4.5.
RequestVersion	<t:RequestServerVersion>	This part specifies the schema version for the **MoveFolder** operation request. The <RequestServerVersion> element is specified in [MS-OXWSCDATA] section 2.2.4.7.

3.1.4.7.1.2 tns:MoveFolderSoapOut Message

The **MoveFolderSoapOut** WSDL message specifies the **MoveFolder** operation response.

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

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

Part name	Element/type	Description
MoveFolderResult	<tns:MoveFolderResponse>	Specifies the response message.
ServerVersion	<t:ServerVersionInfo>	Specifies the server version for the response. The <ServerVersionInfo> element is specified in [MS-OXWSCDATA] section 2.2.4.8.

3.1.4.7.2 Elements

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

Element	Description
<MoveFolder>	Defines a request to move a folder in the server store.
<MoveFolderRespeonse>	Defines a response to a **MoveFolder** operation request.

3.1.4.7.2.1 MoveFolder Element

The <MoveFolder> element defines a request to move a folder in the server store.

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

3.1.4.7.2.2 MoveFolderResponse Element

The <MoveFolderResponse> element represents a response to a **MoveFolder** operation request.

```
<xs:element name="MoveFolderResponse"
  type="m:MoveFolderResponseType"
 />
```

3.1.4.7.3 Complex Types

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

Type	Description
MoveFolderResponseType	Represents a response to a **MoveFolder** operation.
MoveFolderType	Represents an operation to move a folder in a mailbox.

3.1.4.7.3.1 m:MoveFolderResponseType Complex Type

The **MoveFolderResponseType** complex type represents a response to a **MoveFolder** operation request. The **MoveFolderResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.3.15).

```
<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 represents an operation to move folders in a mailbox. The **MoveFolderType** complex type extends the **BaseMoveCopyFolderType** as specified in section [2.2.4.2](#).

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

3.1.4.8 UpdateFolder Operation

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

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

Request

Message Format	Description
tns:UpdateFolderSoapIn	Specifies the SOAP message that requests the properties of an existing item in the server store.

Response

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

3.1.4.8.1 Messages

The following WSDL message definitions are specific to this operation.

Message	Description
UpdateFolderSoapIn	Specifies the request that updates a folder.
UpdateFolderSoapOut	Specifies the response from a request to update a folder.

3.1.4.8.1.1 tns:UpdateFolderSoapIn Message

The **UpdateFolderSoapIn** WSDL message specifies the **UpdateFolder** operation request.

```
<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 five parts of the **UpdateFolderSoapIn** message are described in the following table.

Part Name	Element/Type	Description
request	<tns:UpdateFolder>	Specifies the request message.
Impersonation	<t:ExchangeImpersonation>	Specifies the identifier of the account to impersonate. The <ExchangeImpersonation> element is specified in [MS-OXWSCDATA] section 2.2.4.2.
MailboxCulture	<t:MailboxCulture>	Specifies the culture to use for accessing the server. The cultures are defined by [RFC3066]. The <MailboxCulture> element is specified in [MS-OXWSCDATA] section 2.2.4.5.
RequestVersion	<t:RequestServerVersion>	Specifies the schema version for the **UpdateFolder** operation request. The <RequestServerVersion> element is specified in [MS-OXWSCDATA] section 2.2.4.7.
TimeZoneContext	<t:TimeZoneContext>	Specifies a time zone definition and enables the association of SOAP attributes with the definition. The <TimeZoneContext> element is specified in [MS-OXWSGTZ] section 2.2.4.4.

3.1.4.8.1.2   tns:UpdateFolderSoapOut Message

The **UpdateFolderSoapOut** WSDL message specifies the **UpdateFolder** operation response.

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

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

Part name	Element/type	Description
UpdateFolderResult	<tns:UpdateFolderResponse>	Specifies the response message.
ServerVersion	<t:ServerVersionInfo>	Specifies the server version for the response. The <ServerVersionInfo> element is specified in [MS-OXWSCDATA] section 2.2.4.8.

3.1.4.8.2   Elements

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

Element	Description
<UpdateFolder>	Defines a request to update a folder in the server store.
<UpdateFolderResponse>	Defines a response to an **UpdateFolder** operation request.

3.1.4.8.2.1   UpdateFolder Element

The <UpdateFolder> element defines a request to update a folder in the server store.

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

3.1.4.8.2.2   UpdateFolderResponse Element

The <UpdateFolderResponse> element represents a response to an **UpdateFolder** operation request.

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

3.1.4.8.3   Complex Types

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

Type	Description
UpdateFolderResponseType	Represents a response to an **UpdateFolder** operation.
UpdateFolderType	Represents a request to update folders in a mailbox.
AppendToFolderFieldType	Do not use. This complex type is not implemented.
DeleteFolderFieldType	Represents an **UpdateFolder** operation to delete a property from a folder.
NonEmptyArrayOfFolderChangeDescriptionsType	An array of **ChangeDescriptionType** complex types ([MS-OXWSCORE] section 3.1.4.7.2.4). This array has at least one member.
NonEmptyArrayOfFolderChangesType	An array of **FolderChangeType** complex types (section 2.2.4.8). This array has at least one member.

Type	Description
SetFolderFieldType	Represents 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 represents a response to an **UpdateFolder** operation. The **UpdateFolderResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.3.15).

```
<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 represents an operation to update folders in a mailbox.

```
<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>
```

Child Elements

Element	Type	Description
FolderChanges	**t:NonEmptyArrayOfFolderChangesType**	Represents a collection of changes for a folder.

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.

```
<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>
```

Child Elements

Element	Type	Description
Folder	**t:FolderType**	Represents a regular folder in the server database. The **FolderType** complex type is specified in section 2.2.4.10.
CalendarFolder	**t:CalendarFolderType**	Rrepresents a folder that primarily contains calendar items. The **CalendarFolderType** complex type is specified in [MS-OXWSMTGS] section 2.2.3.3.
ContactsFolder	**t:ContactsFolderType**	Represents a Contacts folder in a mailbox. The **ContactsFolderType** complex type is specified in [MS-OXWSCONT] section 3.1.4.1.2.2.
SearchFolder	**t:SearchFolderType**	Represents a search folder that is contained in a mailbox. The **SearchFolderType** complex type is specified in [MS-OXWSSRCH] section 2.2.3.26.
TasksFolder	**t:TasksFolderType**	Represents a Tasks folder that is contained in a mailbox. The **TasksFolderType** complex type is specified in [MS-OXWSTASK] section 2.2.3.2.

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.

```
<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 is an array of
ChangeDescriptionType complex types ([MS-OXWSCORE] section 3.1.4.7.2.4). This array has at
least one member. The **FolderChangeDescriptionType** (section 2.2.4.7) members describe a
change to a single folder property.

```
    <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>
```

Child Elements

Element	Type	Description
AppendToFolderField	**t:AppendToFolderFieldType**	Do not use. This complex type is not implemented.
SetFolderField	**t:SetFolderFieldType**	Represents an **UpdateFolder** operation to set a property on an existing folder.
DeleteFolderField	**t:DeleteFolderFieldType**	Represents an **UpdateFolder** operation to delete a property from a folder.

3.1.4.8.3.6   t:NonEmptyArrayOfFolderChangesType Complex Type

The **NonEmptyArrayOfFolderChangesType** complex type represents an array of
FolderChangeType complex types (section 2.2.4.8). 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>
```

Child Elements

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

In the **UpdateFolder** example (section 4.6), the folder request XML shows a **FolderChanges** type with a **FolderChangeType** complex type. The folder identifier is identified and the changes are shown in the <Updates> element.

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.

```
  <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>
```

Child Elements

Element	Type	Description
Folder	**t:FolderType**	Represents a regular folder in the server database. The

Element	Type	Description
		FolderType complex type is specified in section 2.2.4.10.
CalendarFolder	**t:CalendarFolderType**	Represents a folder that primarily contains calendar items. The **CalendarFolderType** complex type is specified in [MS-OXWSMTGS] section 2.2.3.3.
ContactsFolder	**t:ContactsFolderType**	Represents a Contacts folder in a mailbox. The **ContactsFolderType** complex type is specified in [MS-OXWSCONT] section 3.1.4.1.2.2.
SearchFolder	**t:SearchFolderType**	Represents a search folder that is contained in a mailbox. The **SearchFolderType** complex type is specified in [MS-OXWSSRCH] section 2.2.3.26.
TasksFolder	**t:TasksFolderType**	Represents a Tasks folder that is contained in a mailbox. The **TasksFolderType** complex type is specified in [MS-OXWSTASK] section 2.2.3.2.

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.

- **CopyFolder** operation example (section 4.1)

- **CreateFolder** operation example (section 4.2)

- **DeleteFolder** operation example (section 4.3)

- **EmptyFolder** operation example (section 4.4)

- **MoveFolder** operation example (section 4.5)

- **UpdateFolder** operation example (section 4.6)

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> **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> **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 following 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.

```
<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> **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> **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 that is specified in this document. The contents of each file are included in this section.

File name	Description	Section
MS-OXWSFOLD.wsdl	Contains the WSDL for the implementation of this protocol.	6.1
MS-OXWSFOLD-types.xsd	Contains the XML schema type definitions that are used in this protocol.	6.2
MS-OXWSFOLD-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	6.3

These files have to be placed in a common folder 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.

6.1 WSDL

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

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

File name	Section
MS-OXWSFOLD-messages.xsd.	6.3

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
    <wsdl:types>
        <xs:schema id="messages" elementFormDefault="qualified" version="Exchange2010"
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
            <xs:include schemaLocation="MS-OXWSFOLD-messages.xsd"/>
            <!-- Add global elements and types from messages.xsd -->
        </xs:schema>
        <xs:schema id="types" elementFormDefault="qualified" version="Exchange2010"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
            <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
            <!-- Add global elements and types from types.xsd -->
        </xs:schema>
```

```
        </wsdl:types>
    <wsdl:portType name="ExchangeServicePortType">
        <wsdl:operation name="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: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: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>
```

6.2 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.

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

Defining specification	File name
[MS-OXWSXPROP], section 6.1	MS-OXWSXPROP-types.xsd

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
elementFormDefault="qualified" version="Exchange2010" id="types">
    <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
    <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
    <xs:include schemaLocation="MS-OXWSXPROP-types.xsd"/>
    <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: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: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="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: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: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:choice>
        </xs:complexType>
    </xs:schema>
```

6.3   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 need to be in the folder that contains the WSDL, types schema, and messages schema files for this protocol.

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

File name	Defining specification
MS-OXWSFOLD-types.xsd	Section 6.2

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
elementFormDefault="qualified" version="Exchange2010" id="messages">

    <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="MS-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   Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products:

- Microsoft® Exchange Server 2007

- Microsoft® Exchange Server 2010

- Microsoft® Exchange Server 2010 Service Pack 1 (SP1)

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified. 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 prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that product does not follow the prescription.

<1> Section 3.1.4.5: Exchange 2007 and the initial release version of Exchange 2010 do not include the **EmptyFolder** operation.

8   Change Tracking

This section identifies changes that were made to the [MS-OXWSFOLD] protocol document between the August 2010 and November 2010 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.

- An extensive rewrite, addition, or deletion of major portions of content.

- Changes made for template compliance.

- 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 language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

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

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

- New content added.

- Content updated.

- Content removed.

- New product behavior note added.

- Product behavior note updated.

- Product behavior note removed.

- New protocol syntax added.

- Protocol syntax updated.

- Protocol syntax removed.

- New content added due to protocol revision.

- Content updated due to protocol revision.

- Content removed due to protocol revision.

- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.

- Protocol syntax removed due to protocol revision.

- New content added for template compliance.

- Content updated for template compliance.

- Content removed for template compliance.

- Obsolete document removed.

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

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

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
1.2.1 Normative References	Added [MS-OXDSCLI] to the list of normative references.	Y	Content updated.
3.1.4.1.1.1 tns:CopyFolderSoapIn Message	57838 Updated the introductory text and added WSDL syntax.	N	Content updated.
3.1.4.1.1.2 tns:CopyFolderSoapOut Message	53989 Added information about SOAP content.	N	Content updated.
3.1.4.2.1.1 tns:CreateFolderSoapIn Message	57838 Updated the introductory text and added WSDL syntax.	N	Content updated.
3.1.4.3.1.1 tns:CreateManagedFolderSoapIn Message	57838 Updated the introductory text and added the WSDL syntax.	N	Content updated.
3.1.4.3.1.2 tns:CreateManagedFolderSoapOut Message	53989 Added information about SOAP content.	N	Content updated.
3.1.4.4.1.1 tns:DeleteFolderSoapIn Message	57838 Updated the introductory text and added the WSDL syntax.	N	Content updated.
3.1.4.4.1.2 tns:DeleteFolderSoapOut Message	53989 Added information about SOAP	Y	New content added.

Release: Tuesday, October 26, 2010

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
	content.		
3.1.4.5 EmptyFolder Operation	57838 Added Exchange 2007 to the list of products that do not include the EmptyFolder operation.	Y	Product behavior note updated.
3.1.4.5.1.1 tns:EmptyFolderSoapIn Message	57838 Updated the introductory text and added the WSDL syntax.	N	Content updated.
3.1.4.5.1.2 tns:EmptyFolderSoapOut Message	53989 Added information about SOAP content.	Y	New content added.
3.1.4.6.1.1 tns:GetFolderSoapIn Message	57838 Updated the introductory text and added the WSDL syntax.	N	Content updated.
3.1.4.6.1.2 tns:GetFolderSoapOut Message	53989 Added information about SOAP content.	Y	New content added.
3.1.4.7.1.1 tns:MoveFolderSoapIn Message	57838 Updated the introductory text and added WSDL Syntax.	N	Content updated.
3.1.4.8.1.1 tns:UpdateFolderSoapIn Message	57838 Updated the introductory text and added the WSDL syntax.	N	Content updated.
7 Appendix B: Product Behavior	57838 Added Exchange 2007 to the list of applicable products.	Y	New product behavior note added.

Release: Tuesday, October 26, 2010

9 Index