Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.

- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.

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

- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](https://aka.ms/opensp) or the [Microsoft Community Promise](https://aka.ms/mcpp). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).

- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](https://aka.ms/opensp).

- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](https://www.microsoft.com/trademarks).

- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>7/16/2012</td>
<td>0.1</td>
<td>New</td>
<td>Released new document.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>1.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>1.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>7/26/2013</td>
<td>1.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>1.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>1.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>1.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>1.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>1.3</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>3/16/2015</td>
<td>2.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>5/26/2015</td>
<td>2.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/14/2015</td>
<td>3.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>6/13/2016</td>
<td>4.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>4.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/24/2018</td>
<td>5.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/1/2018</td>
<td>6.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/22/2021</td>
<td>7.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>8/17/2021</td>
<td>8.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>2/15/2022</td>
<td>8.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/16/2024</td>
<td>9.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
</tbody>
</table>
Table of Contents

1 Introduction ........................................................................................................... 5
  1.1 Glossary ........................................................................................................... 5
  1.2 References ....................................................................................................... 6
    1.2.1 Normative References ............................................................................... 6
    1.2.2 Informative References ........................................................................... 7
  1.3 Overview ......................................................................................................... 7
  1.4 Relationship to Other Protocols ..................................................................... 8
  1.5 Prerequisites/Preconditions ........................................................................... 8
  1.6 Applicability Statement ................................................................................... 8
  1.7 Versioning and Capability Negotiation ......................................................... 8
  1.8 Vendor-Extensible Fields ................................................................................ 9
  1.9 Standards Assignments ................................................................................... 9

2 Messages............................................................................................................ 10
  2.1 Transport ....................................................................................................... 10
  2.2 Common Message Syntax .............................................................................. 10
    2.2.1 Namespaces ............................................................................................. 10
    2.2.2 Messages ................................................................................................ 10
    2.2.3 Elements .................................................................................................. 10
    2.2.4 Complex Types ...................................................................................... 11
    2.2.5 Simple Types .......................................................................................... 11
    2.2.6 Attributes ............................................................................................... 11
    2.2.7 Groups ................................................................................................... 11
    2.2.8 Attribute Groups .................................................................................... 11

3 Protocol Details ................................................................................................ 12
  3.1 ExchangeServicePortType Server Details ................................................... 12
    3.1.1 Abstract Data Model ............................................................................... 12
    3.1.2 Timers ..................................................................................................... 12
    3.1.3 Initialization ............................................................................................. 12
    3.1.4 Message Processing Events and Sequencing Rules ............................... 12
      3.1.4.1 ArchiveItem ....................................................................................... 12
        3.1.4.1.1 Messages .................................................................................... 13
          3.1.4.1.1.1 ArchiveItemSoapIn ............................................................. 13
          3.1.4.1.1.2 ArchiveItemSoapOut ......................................................... 14
        3.1.4.1.2 Elements ..................................................................................... 15
          3.1.4.1.2.1 ArchiveItem ................................................................. 15
          3.1.4.1.2.2 ArchiveItemResponse .................................................... 15
        3.1.4.1.3 Complex Types ......................................................................... 15
          3.1.4.1.3.1 ArchiveItemType ............................................................ 15
          3.1.4.1.3.2 ArchiveItemResponseType ........................................... 16
        3.1.4.1.4 Simple Types .......................................................................... 16
        3.1.4.1.5 Attributes ................................................................................. 16
        3.1.4.1.6 Groups ....................................................................................... 16
        3.1.4.1.7 Attribute Groups .................................................................... 16
      3.1.4.2 CreateFolderPath .............................................................................. 16
        3.1.4.2.1 Messages .................................................................................... 17
          3.1.4.2.1.1 CreateFolderPathSoapIn .................................................. 17
          3.1.4.2.1.2 CreateFolderPathSoapOut .............................................. 18
        3.1.4.2.2 Elements ..................................................................................... 19
          3.1.4.2.2.1 CreateFolderPath ............................................................ 19
          3.1.4.2.2.2 CreateFolderPathResponse ........................................... 19
        3.1.4.2.3 Complex Types ......................................................................... 19
          3.1.4.2.3.1 CreateFolderPathType .................................................... 19
          3.1.4.2.3.2 CreateFolderPathResponseType .................................. 20
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.4.2.4</td>
<td>Simple Types</td>
<td>20</td>
</tr>
<tr>
<td>3.1.4.2.5</td>
<td>Attributes</td>
<td>20</td>
</tr>
<tr>
<td>3.1.4.2.6</td>
<td>Groups</td>
<td>20</td>
</tr>
<tr>
<td>3.1.4.2.7</td>
<td>Attribute Groups</td>
<td>20</td>
</tr>
<tr>
<td>3.1.5</td>
<td>Timer Events</td>
<td>20</td>
</tr>
<tr>
<td>3.1.6</td>
<td>Other Local Events</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>Protocol Examples</td>
<td>21</td>
</tr>
<tr>
<td>4.1</td>
<td>Archiving an Item</td>
<td>21</td>
</tr>
<tr>
<td>4.2</td>
<td>Creating a Folder Path</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>Security</td>
<td>24</td>
</tr>
<tr>
<td>5.1</td>
<td>Security Considerations for Implementers</td>
<td>24</td>
</tr>
<tr>
<td>5.2</td>
<td>Index of Security Parameters</td>
<td>24</td>
</tr>
<tr>
<td>6</td>
<td>Appendix A: Full WSDL</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>Appendix B: Full XML Schema</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>Appendix C: Product Behavior</td>
<td>28</td>
</tr>
<tr>
<td>9</td>
<td>Change Tracking</td>
<td>29</td>
</tr>
<tr>
<td>10</td>
<td>Index</td>
<td>30</td>
</tr>
</tbody>
</table>
1 Introduction

The Archiving Web Service Protocol enables clients to use a web service to archive items in a mailbox.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

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

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

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

Inbox folder: A special folder that is the default location for Message objects received by a user or resource.

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

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

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

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

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

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

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

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

web service: A unit of application logic that provides data and services to other applications and can be called by using standard Internet transport protocols such as HTTP, Simple Mail Transfer...
Protocol (SMTP), or File Transfer Protocol (FTP). Web services can perform functions that range from simple requests to complicated business processes.

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

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

**WSDL operation**: A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

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

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

**XML namespace prefix**: An abbreviated form of an XML namespace, as described in [XML].

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

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

### 1.2 References

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

**1.2.1 Normative References**

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

- [MS-OXWSCDATA] Microsoft Corporation, "Common Web Service Data Types".
- [MS-OXWSCORE] Microsoft Corporation, "Core Items Web Service Protocol".
- [MS-OXWSFOLD] Microsoft Corporation, "Folders and Folder Permissions Web Service Protocol".
- [MS-OXWSGTZ] Microsoft Corporation, "Get Server Time Zone Web Service Protocol".
1.3 Overview

Archive **mailboxes** are secondary mailboxes that are associated with a user. Archive mailboxes are typically used to manage email storage limits. For example, older email items might periodically be moved from the **Inbox folder** to the archive mailbox.

This protocol provides operations to archive a set of mail items from a primary mailbox. Archiving items in the Inbox folder in this way preserves the folder hierarchy of the items. In addition, archive mailboxes can be stored either locally on a client or remotely, in a way that is mostly opaque to a user, by using a folder path to point to the contents of the archive.
1.4 Relationship to Other Protocols

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

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

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

![Layering Diagram](image)

**Figure 1: This protocol in relation to other protocols**

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

1.5 Prerequisites/Preconditions

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

1.6 Applicability Statement

This protocol is applicable to environments that use web services to implement archiving from a user's mailbox. This protocol is applicable to all SOAP-based clients, as described in [SOAP1.1].

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses SOAP 1.1, as described in section 2.1.
- **Protocol Versions:** This protocol describes only one WSDL port type version. The `t:RequestServerVersion` element, as described in [MS-OXWSCDATA] section 2.2.3.9, identifies the WSDL port type version of the request. The `t:ServerVersionInfo` element, as described in [MS-OXWSCDATA] section 2.2.3.10 identifies the version of the server responding to the request.
- **Security and Authentication Methods:** This protocol relies on the web server that is hosting it to perform authentication.
- **Localization**: This protocol uses the `MailboxCulture` element, as described in [MS-OXWSCDATA] section 2.2.4.43, to specify the culture of a mailbox.

- **Capability Negotiation**: This protocol does not support version negotiation.

1.8 **Vendor-Extensible Fields**

None.

1.9 **Standards Assignments**

None.
2 Messages

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

2.1 Transport

This protocol uses SOAP 1.1, as specified in [SOAP1.1].

The protocol MUST support SOAP over HTTP, as specified in [RFC2616]. The protocol SHOULD use secure communications by means of HTTPS, as specified in [RFC2818].

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses XML schema, as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and WSDL, as specified in [WSDL].

2.2.1 Namespaces

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

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>soap</td>
<td><a href="http://schemas.xmlsoap.org/wsd1/soap/">http://schemas.xmlsoap.org/wsd1/soap/</a></td>
<td>[SOAP1.1]</td>
</tr>
<tr>
<td>t</td>
<td><a href="http://schemas.microsoft.com/exchange/services/2006/types">http://schemas.microsoft.com/exchange/services/2006/types</a></td>
<td></td>
</tr>
<tr>
<td>tns</td>
<td><a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a></td>
<td></td>
</tr>
<tr>
<td>wsd1</td>
<td><a href="http://schemas.xmlsoap.org/wsd1/">http://schemas.xmlsoap.org/wsd1/</a></td>
<td>[WSDL]</td>
</tr>
<tr>
<td>wsi</td>
<td><a href="http://ws-i.org/schemas/conformanceClaim/">http://ws-i.org/schemas/conformanceClaim/</a></td>
<td>[WSIBASIC]</td>
</tr>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>[XMLSCHEMA1/2]</td>
</tr>
</tbody>
</table>

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

This specification does not define any common XML schema element definitions.
2.2.4 Complex Types
This specification does not define any common XML schema complex type definitions.

2.2.5 Simple Types
This specification does not define any common XML schema simple type definitions.

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.

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

3.1 ExchangeServicePortType Server Details

This protocol defines a single WSDL port type with two operations. These operations enable clients to archive items in a mailbox and create a path to the archive location.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArchiveItem</td>
<td>The ArchiveItem operation moves items from the primary mailbox to the archive mailbox.</td>
</tr>
<tr>
<td>CreateFolderPath</td>
<td>The CreateFolderPath operation creates a Uniform Resource Identifier (URI) that points to the storage location for the archive mailbox.</td>
</tr>
</tbody>
</table>

3.1.4.1 ArchiveItem

The ArchiveItem operation moves items from the primary mailbox to the archive mailbox.

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

```
<wSDL:operation name="ArchiveItem" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/">
 <wSDL:input message="tns:ArchiveItemSoapIn"/>
 <wSDL:output message="tns:ArchiveItemSoapOut"/>
</wSDL:operation>
```
The following is the WSDL binding specification of the `ArchiveItem` operation.

```xml
<wsdl:operation name="ArchiveItem" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <soap:operation
    soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/ArchiveItem"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <wsdl:input>
      <soap:header message="tns:ArchiveItemSoapIn" part="Impersonation" use="literal"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
      </soap:header>
      <soap:header message="tns:ArchiveItemSoapIn" part="MailboxCulture" use="literal"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
      </soap:header>
      <soap:header message="tns:ArchiveItemSoapIn" part="RequestVersion" use="literal"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
      </soap:header>
      <soap:body use="literal" parts="request"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
      </soap:body>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" parts="ArchiveItemResult"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
      </soap:body>
      <soap:header message="tns:ArchiveItemSoapOut" part="ServerVersion" use="literal"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
      </soap:header>
    </wsdl:output>
  </soap:operation>
</wsdl:operation>
```

The protocol client sends an `ArchiveItemSoapIn` request **WSDL message**, as specified in section 3.1.4.1.1.1, and the protocol server responds with an `ArchiveItemSoapOut` response message, as specified in section 3.1.4.1.1.2.

### 3.1.4.1.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArchiveItemSoapIn</td>
<td>Specifies the request to archive items.</td>
</tr>
<tr>
<td>ArchiveItemSoapOut</td>
<td>Specifies the response to the request to archive items.</td>
</tr>
</tbody>
</table>

### 3.1.4.1.1.1 ArchiveItemSoapIn

The **ArchiveItemSoapIn** **WSDL message** specifies the request to archive items.

The following is the **ArchiveItemSoapIn** **WSDL message** specification.

```xml
<wsdl:message name="ArchiveItemSoapIn" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:part name="request" element="tns:ArchiveItem"/>
  <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
  <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
  <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
```

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

The parts of the **ArchiveItemSoapIn** **WSDL message** are described in the following table.
### Part name | Element/type | Description
--- | --- | ---
request | ArchiveItem (section 3.1.4.1.2.1) | Specifies the SOAP body of the request to archive items.

**Impersonation**

ExchangeImpersonation ([MS-OXWSDATA] section 2.2.3.3)

Specifies a SOAP header that identifies the user whom the client is impersonating.

**MailboxCulture**

MailboxCulture ([MS-OXWSDATA] section 2.2.4.43)

Specifies a SOAP header that identifies the culture to use when accessing the mailbox. The cultures are defined in [RFC3066].

**RequestVersion**

RequestServerVersion ([MS-OXWSDATA] section 2.2.3.9)

Specifies a SOAP header that identifies the schema version for the ArchiveItem operation request.

### 3.1.4.1.1.2 ArchiveItemSoapOut

The **ArchiveItemSoapOut WSDL message** specifies the response to the ArchiveItem operation request.

The following is the **ArchiveItemSoapOut WSDL message** specification.

```xml
<wsdl:message name="ArchiveItemSoapOut" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:part name="ArchiveItemResult" element="tns:ArchiveItemResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

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

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

<table>
<thead>
<tr>
<th>Part name</th>
<th>Element/type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArchiveItemResult</td>
<td>ArchiveItemResponse (section 3.1.4.1.2.2)</td>
<td>Specifies the SOAP body of the response to an ArchiveItem operation request.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>ServerVersionInfo ([MS-OXWSDATA] section 2.2.3.10)</td>
<td>Specifies a SOAP header that identifies the server version for the response.</td>
</tr>
</tbody>
</table>

A successful ArchiveItem operation request returns an **ArchiveItemResponse** element, as specified in section 3.1.4.1.3.2, with the **ResponseClass** attribute of the ArchiveItemResponseMessage element set to "Success". The **ResponseCode** element of the ArchiveItemResponseMessage element is set to "NoError". The **ArchiveItemResponseMessage** element is a child element of the **ArrayOfResponseMessagesType** complex type, as specified in [MS-OXWSDATA] section 2.2.4.12.

If the ArchiveItem operation request is not successful, it returns an **ArchiveItemResponse** element with the **ResponseClass** attribute of the ArchiveItemResponseMessage element set to "Error". The **ResponseCode** element of the ArchiveItemResponseMessage element is set to one of the common errors defined in [MS-OXWSDATA] section 2.2.5.24.
3.1.4.1.2 Elements
The following table summarizes the XML schema element definitions that are specific to this operation.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArchiveItem</td>
<td>The input data for the ArchiveItem WSDL operation.</td>
</tr>
<tr>
<td>ArchiveItemResponse</td>
<td>The result data for the ArchiveItem WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.1.2.1 ArchiveItem
The ArchiveItem element specifies the input data for the ArchiveItem WSDL operation.

```xml
<xs:element name="ArchiveItem" type="tns:ArchiveItemType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

3.1.4.1.2.2 ArchiveItemResponse
The ArchiveItemResponse element specifies the result data for the ArchiveItem WSDL operation.

```xml
<xs:element name="ArchiveItemResponse" type="tns:ArchiveItemResponseType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArchiveItemResponseType</td>
<td>Specifies the response to the ArchiveItem operation (section 3.1.4.1).</td>
</tr>
<tr>
<td>ArchiveItemType</td>
<td>Specifies a request to move items from the primary mailbox to an archive folder.</td>
</tr>
</tbody>
</table>

3.1.4.1.3.1 ArchiveItemType
Namespace: http://schemas.microsoft.com/exchange/services/2006/messages
The ArchiveItemType complex type specifies a request to move items from the primary mailbox to an archive folder. This type extends the BaseRequestType complex type, as described in [MS-OXWSDATA] section 2.2.4.17.

```xml
<xs:complexType name="ArchiveItemType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestType">
      <xs:sequence>
        <xs:element name="ArchiveSourceFolderId" type="t:TargetFolderIdType"/>
        <xs:element name="ItemIds" type="t:NonEmptyArrayOfBaseItemIdsType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```
archiveSourceFolderId: An element of type TargetFolderIdType, as defined in [MS-OXWSFOLD] section 2.2.4.16, that specifies the identifier of the source folder.

ItemIds: An element of type NonEmptyArrayOfBaseItemIdsType, as defined in [MS-OXWSCORE] section 2.2.4.31, that specifies the item identifiers.

3.1.4.1.3.2 ArchiveItemResponseType

Namespace: http://schemas.microsoft.com/exchange/services/2006/messages

The ArchiveItemResponseType complex type specifies the response for an ArchiveItem operation. This type extends the BaseResponseMessageType complex type, as described in [MS-OXWSCDATA] section 2.2.4.18.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 CreateFolderPath

The CreateFolderPath operation creates a Uniform Resource Identifier (URI) that points to the storage location for the archive mailbox.

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

```xml
<wsdl:operation name="CreateFolderPath" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input message="tns:CreateFolderPathSoapIn"/>
    <wsdl:output message="tns:CreateFolderPathSoapOut"/>
</wsdl:operation>
```

The following is the WSDL binding specification of the CreateFolderPath operation.

```xml
<wsdl:operation name="CreateFolderPath" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
```
The protocol client sends a CreateFolderPathSoapIn request **WSDL message**, as specified in section 3.1.4.2.1.1, and the protocol server responds with a CreateFolderPathSoapOut response message, as specified in section 3.1.4.2.1.2.

### 3.1.4.2.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateFolderPathSoapIn</td>
<td>Specifies the request to create a folder path.</td>
</tr>
<tr>
<td>CreateFolderPathSoapOut</td>
<td>Specifies the response to the CreateFolderPathSoapIn request.</td>
</tr>
</tbody>
</table>

#### 3.1.4.2.1.1 CreateFolderPathSoapIn

The CreateFolderPathSoapIn **WSDL message** specifies the request to create a folder path.

The following is the CreateFolderPathSoapIn **WSDL message** specification.

```
<wsdl:message name="CreateFolderPathSoapIn" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:part name="request" element="tns:CreateFolderPath"/>
  <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 CreateFolderPathSoapIn WSDL message is the input message for the **SOAP action** [http://schemas.microsoft.com/exchange/services/2006/messages/CreateFolderPath](http://schemas.microsoft.com/exchange/services/2006/messages/CreateFolderPath).

The parts of the CreateFolderPathSoapIn WSDL message are described in the following table.
<table>
<thead>
<tr>
<th>Part name</th>
<th>Element/type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>CreateFolderPath  (section 3.1.4.2.2.1)</td>
<td>Specifies the SOAP body of the request to create a folder path.</td>
</tr>
<tr>
<td>Impersonation</td>
<td>ExchangeImpersonation  ([MS-OXWSCDATA] section 2.2.3.3)</td>
<td>Specifies a SOAP header that identifies the user whom the client is impersonating.</td>
</tr>
<tr>
<td>MailboxCulture</td>
<td>MailboxCulture  ([MS-OXWSCDATA] section 2.2.4.43)</td>
<td>Specifies a SOAP header that identifies the culture to use to access the mailbox. The cultures are defined in [RFC3066].</td>
</tr>
<tr>
<td>RequestVersion</td>
<td>RequestServerVersion  ([MS-OXWSCDATA] section 2.2.3.9)</td>
<td>Specifies a SOAP header that identifies the schema version for the CreateFolderPath operation request.</td>
</tr>
<tr>
<td>TimeZoneContext</td>
<td>TimeZoneContext  ([MS-OXWSGTZ] section 2.2.3.4)</td>
<td>Specifies a SOAP header that identifies the time zone to use for all responses from the server. All times that are returned from the server will be converted to the specified time zone.</td>
</tr>
</tbody>
</table>

### 3.1.4.2.1.2 CreateFolderPathSoapOut

The CreateFolderPathSoapOut WSDL message specifies the response to the CreateFolderPathSoapIn request.

The following is the CreateFolderPathSoapOut WSDL message specification.

```xml
<wSDL:message name="CreateFolderPathSoapOut" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/">
  <wSDL:part name="CreateFolderPathResult" element="tns:CreateFolderPathResponse"/>
  <wSDL:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wSDL:message>
```

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

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

<table>
<thead>
<tr>
<th>Part name</th>
<th>Element/type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateFolderPathResult</td>
<td>CreateFolderPathResponse (section 3.1.4.2.2.2)</td>
<td>Represents the result of this operation.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>ServerVersionInfo  ([MS-OXWSCDATA] section 2.2.3.10)</td>
<td>Specifies a SOAP header that identifies the server version for the response.</td>
</tr>
</tbody>
</table>

A successful CreateFolderPath operation request returns a CreateFolderPathResponse element with the ResponseClass attribute of the CreateFolderPathResponseMessage element set to "Success". The ResponseCode element of the CreateFolderPathResponseMessage element is set to "NoError". The CreateFolderPathResponseMessage element is a child element of the ArrayOfResponseMessagesType complex type, as specified in [MS-OXWSCDATA] section 2.2.4.12.
If the `CreateFolderPath` operation request is not successful, it returns a `CreateFolderPathResponse` element with the `ResponseClass` attribute of the `CreateFolderPathResponseMessage` element set to "Error". The `ResponseCode` element of the `CreateFolderPathResponseMessage` element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

### 3.1.4.2.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateFolderPath</td>
<td>The input data for the <code>CreateFolderPath</code> WSDL operation.</td>
</tr>
<tr>
<td>CreateFolderPathResponse</td>
<td>The result data for the <code>CreateFolderPath</code> WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.2.2.1 CreateFolderPath

The `CreateFolderPath` element specifies the input data for the `CreateFolderPath` WSDL operation.

```xml
<xs:element name="CreateFolderPath" type="tns:CreateFolderPathType"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

### 3.1.4.2.2.2 CreateFolderPathResponse

The `CreateFolderPathResponse` element specifies the result data for the `CreateFolderPath` WSDL operation.

```xml
<xs:element name="CreateFolderPathResponse" type="tns:CreateFolderPathResponseType"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

### 3.1.4.2.3 Complex Types

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateFolderPathResponseType</td>
<td>Specifies the response for the <code>CreateFolderPath</code> operation (section 3.1.4.2).</td>
</tr>
<tr>
<td>CreateFolderPathType</td>
<td>Specifies a request to create a folder path.</td>
</tr>
</tbody>
</table>

### 3.1.4.2.3.1 CreateFolderPathType

**Namespace:** http://schemas.microsoft.com/exchange/services/2006/messages

The `CreateFolderPathType` complex type specifies a request to create a folder path. This type extends the `BaseRequestType` complex type, as described in [MS-OXWSCDATA] section 2.2.4.17.
<xs:complexType name="CreateFolderPathType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:extension base="tns:BaseRequestType">
    <xs:sequence>
      <xs:element name="ParentFolderId" type="t:TargetFolderIdType"/>
      <xs:element name="RelativeFolderPath" type="t:NonEmptyArrayOfFoldersType"/>
    </xs:sequence>
  </xs:extension>
</xs:complexType>

**ParentFolderPath:** An element of type **TargetFolderIdType**, as defined in [MS-OXWSFOLD] section 2.2.4.16, that specifies the parent folder identifier.

**RelativeFolderPath:** An element of type **NonEmptyArrayOfFoldersType**, as defined in [MS-OXWSFOLD] section 3.1.4.2.3.3, that specifies the relative folder path.

### 3.1.4.2.3.2 CreateFolderPathResponseType

**Namespace:** http://schemas.microsoft.com/exchange/services/2006/messages

The **CreateFolderPathResponseType** complex type specifies the response for the **CreateFolderPath** operation. This type extends the **BaseResponseMessageType** complex type, as described in [MS-OXWSDATA] section 2.2.4.18.

```xml
<xs:complexType name="CreateFolderPathResponseType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
```

### 3.1.4.2.4 Simple Types

None.

### 3.1.4.2.5 Attributes

None.

### 3.1.4.2.6 Groups

None.

### 3.1.4.2.7 Attribute Groups

None.

### 3.1.5 Timer Events

None.

### 3.1.6 Other Local Events

None.
4 Protocol Examples

4.1 Archiving an Item

This example of the ArchiveItem operation, as defined in section 3.1.4.1, shows how a client can archive items from the Inbox folder.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
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="Exchange2013"/>
  </soap:Header>
  <soap:Body>
    <m:ArchiveItem>
      <m:ArchiveSourceFolderId>
        <t:DistinguishedFolderId Id="inbox"/>
      </m:ArchiveSourceFolderId>
      <m:ItemIds>
        <t:ItemId Id="AQMkAGVmNDAyODMwLTNkMzMALTRhMGQtYTRlOS1kODU3NjkwMkE2ZGQARgAAA2L1bNf8ooZNtO4guk3auRYHALkzPVMoYjIjKEpkWQ0cX0AAAMOAAAauTM+8yglUWNauRG5BBwrQAAAxoAAAA=" ChangeKey="CQAAABYAAAC5Mz7zKCNNRY1q5EbhEHyCtAAAAAB7"/>
      </m:ItemIds>
    </m:ArchiveItem>
  </soap:Body>
</soap:Envelope>
```

The server constructs the following successful response to the request and sends it to the client.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types">
  <Header/>
  <Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <m:ArchiveItemResponse xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
      <m:ResponseMessages>
        <m:ArchiveItemResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
        </m:ArchiveItemResponseMessage>
      </m:ResponseMessages>
    </m:ArchiveItemResponse>
  </Body>
</envelope>
```
4.2 Creating a Folder Path

This example of the `CreateFolderPath` operation, as defined in section 3.1.4.2, shows how the client creates a folder path. This example creates three folders in the Inbox folder. Each folder is a child of the folder that precedes it.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope 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="Exchange2013"/>
    <t:MailboxCulture>en-US</t:MailboxCulture>
    <t:TimeZoneContext>
        <t:TimeZoneDefinition Id="GMT Standard Time"/>
    </t:TimeZoneContext>
</soap:Header>
<soap:Body>
    <m:CreateFolderPath>
        <m:ParentFolderId>
            <t:DistinguishedFolderId Id="inbox"/>
        </m:ParentFolderId>
        <m:RelativeFolderPath>
            <t:Folder>
                <t:DisplayName>MyFirstLevelFolder</t:DisplayName>
            </t:Folder>
            <t:Folder>
                <t:DisplayName>MySecondLevelFolder</t:DisplayName>
            </t:Folder>
            <t:Folder>
                <t:DisplayName>MyThirdLevelFolder</t:DisplayName>
            </t:Folder>
        </m:RelativeFolderPath>
    </m:CreateFolderPath>
</soap:Body>
</soap:Envelope>
```

The server constructs the following successful response to the request. In this response, the server creates three folders in the Inbox folder.

```xml
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Header>
        <h:ServerVersionInfo MajorVersion="15"
            MinorVersion="0"
            MajorBuildNumber="526"
            MinorBuildNumber="0"
            Version="Exchange2013"
            xmlns:h=" http://schemas.microsoft.com/exchange/services/2006/types"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
    </s:Header>
    <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:s="http://www.w3.org/2001/XMLSchema">
        <m:CreateFolderPathResponse
            xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema">
            <m:ResponseMessages>
                <m:CreateFolderPathResponseMessage ResponseClass="Success">
                    <m:ResponseCode>NoError</m:ResponseCode>
                    <m:Folders>
                        <t:Folder>
                            <t:FolderId Id="AAMkADEzOTExYjJkLTYxZDAtNDgxOC04YzQyLTU0OGY1Yzc3ZGY0MwAuAAAAAADhS2QUsLGoTbY+lhGktZkcAQCYnYF59gJhQaoeGgGqm4QrAAABq6Wxb"/>
                        </t:Folder>
                    </m:Folders>
                </m:CreateFolderPathResponseMessage>
            </m:ResponseMessages>
        </m:CreateFolderPathResponse>
    </s:Body>
</s:Envelope>
```
<t:DisplayName>MyFirstLevelFolder</t:DisplayName>
<t:TotalCount>0</t:TotalCount>
<t:ChildFolderCount>0</t:ChildFolderCount>
<t:UnreadCount>0</t:UnreadCount>
</t:Folder>
</m:Folders>
</m:CreateFolderPathResponseMessage>
<m:CreateFolderPathResponseMessage ResponseClass="Success">
  <m:ResponseCode>NoError</m:ResponseCode>
  <m:Folders>
    <t:Folder>
      <t:FolderId Id="AAMkADEzOTExYjJkLTYxZDAtNDgxOC04YzQyLTIaMGWauAAAAADhS2QUsLGoTbY+lhGktZkcAQCYnYF59gJhQaeoGGgq4qARbq6Wxg"/>
      <t:DisplayName>MySecondLevelFolder</t:DisplayName>
      <t:TotalCount>0</t:TotalCount>
      <t:ChildFolderCount>0</t:ChildFolderCount>
      <t:UnreadCount>0</t:UnreadCount>
    </t:Folder>
  </m:Folders>
</m:CreateFolderPathResponseMessage>
<m:CreateFolderPathResponseMessage ResponseClass="Success">
  <m:ResponseCode>NoError</m:ResponseCode>
  <m:Folders>
    <t:Folder>
      <t:FolderId Id="AAMkADEzOTExYjJkLTYxZDAtNDgxOC04YzQyLTIaMGWauAAAAADhS2QUsLGoTbY+lhGktZkcAQCYnYF59gJhQaeoGGgq4qARbq6Wxg"/>
      <t:DisplayName>MyThirdLevelFolder</t:DisplayName>
      <t:TotalCount>0</t:TotalCount>
      <t:ChildFolderCount>0</t:ChildFolderCount>
      <t:UnreadCount>0</t:UnreadCount>
    </t:Folder>
  </m:Folders>
</m:CreateFolderPathResponseMessage>
</m:ResponseMessages>
</m:CreateFolderPathResponse>
</s:Body>
</s:Envelope>
5 Security

5.1 Security Considerations for Implementers
None.

5.2 Index of Security Parameters
None.
6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided in this appendix.

```xml
<?xml version="1.0" encoding="UTF-8"?>
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
    targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:types>
        <xs:schema elementFormDefault="qualified"
            targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
            version="Exchange2016" id="messages" xmlns:xs="http://www.w3.org/2001/XMLSchema">
            <xs:include schemaLocation="MS-OXWSARCH-messages.xsd"/>
        </xs:schema>
    </wsdl:types>
    <wsdl:portType name="ExchangeServicePortType">
        <wsdl:operation name="ArchiveItem">
            <wsdl:input message="tns:ArchiveItemSoapIn"/>
            <wsdl:output message="tns:ArchiveItemSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="CreateFolderPath">
            <wsdl:input message="tns:CreateFolderPathSoapIn"/>
            <wsdl:output message="tns:CreateFolderPathSoapOut"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="ArchiveItem">
            <soap:operation soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/ArchiveItem"/>
            <soap:input>
                <soap:header message="tns:ArchiveItemSoapIn" part="Impersonation" use="literal"/>
                <soap:header message="tns:ArchiveItemSoapIn" part="MailboxCulture" use="literal"/>
                <soap:header message="tns:ArchiveItemSoapIn" part="RequestVersion" use="literal"/>
                <soap:body use="literal" parts="request"/>
            </soap:input>
            <soap:output>
                <soap:header message="tns:ArchiveItemSoapOut" part="ServerVersion" use="literal"/>
            </soap:output>
        </wsdl:operation>
        <wsdl:operation name="CreateFolderPath">
            <soap:input>
                <soap:header message="tns:CreateFolderPathSoapIn" part="Impersonation" use="literal"/>
                <soap:header message="tns:CreateFolderPathSoapIn" part="MailboxCulture" use="literal"/>
                <soap:header message="tns:CreateFolderPathSoapIn" part="RequestVersion" use="literal"/>
                <soap:header message="tns:CreateFolderPathSoapIn" part="TimeZoneContext" use="literal"/>
                <soap:body use="literal" parts="request"/>
            </soap:input>
            <soap:output>
                <soap:header message="tns:CreateFolderPathSoapOut" part="ServerVersion" use="literal"/>
            </soap:output>
        </wsdl:operation>
    </wsdl:binding>
</wsdl:definitions>
```
<wsdl:binding name="ArchiveItemSoapIn">
  <wsdl:message name="ArchiveItemSoapIn">
    <wsdl:part name="request" element="tns:ArchiveItem"/>
    <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
    <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
  </wsdl:message>
</wsdl:binding>

<wsdl:message name="ArchiveItemSoapOut">
  <wsdl:part name="ArchiveItemResponse" element="tns:ArchiveItemResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>

<wsdl:message name="CreateFolderPathSoapIn">
  <wsdl:part name="request" element="tns:CreateFolderPath"/>
  <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="CreateFolderPathSoapOut">
  <wsdl:part name="CreateFolderPathResponse" element="tns:CreateFolderPathResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
7 Appendix B: Full XML Schema

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

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

<table>
<thead>
<tr>
<th>File name</th>
<th>Defining specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-OXWSCDATA-messages.xsd</td>
<td>[MS-OXWSCDATA] section 7.1</td>
</tr>
<tr>
<td>MS-OXWSFOLD-types.xsd</td>
<td>[MS-OXWSFOLD] section 7.2</td>
</tr>
</tbody>
</table>

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

```xml
<?xml version="1.0" encoding="UTF-8"?>
    xmlns:x="http://schemas.microsoft.com/exchange/services/2006/types"
    elementFormDefault="qualified"
    targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
    version="Exchange2016"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema">
    <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
    <xs:import schemaLocation="MS-OXWSFOLD-types.xsd"
    namespace="http://schemas.microsoft.com/exchange/services/2006/types"/>
    <xs:complexType name="ArchiveItemType">
        <xs:complexContent mixed="false">
            <xs:extension base="tns:BaseRequestType">
                <xs:sequence>
                    <xs:element name="ArchiveSourceFolderId" type="t:TargetFolderIdType"/>
                    <xs:element name="ItemIds" type="t:NonEmptyArrayOfBaseItemIdsType"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="ArchiveItem" type="tns:ArchiveItemType"/>
    <xs:complexType name="ArchiveItemResponseType">
        <xs:complexContent mixed="false">
            <xs:extension base="tns:BaseResponseMessageType"/>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="ArchiveItemResponse" type="tns:ArchiveItemResponseType"/>
    <xs:complexType name="CreateFolderPathType">
        <xs:complexContent mixed="false">
            <xs:extension base="tns:BaseRequestType">
                <xs:sequence>
                    <xs:element name="ParentFolderId" type="t:TargetFolderIdType"/>
                    <xs:element name="RelativeFolderPath" type="t:NonEmptyArrayOfFoldersType"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="CreateFolderPath" type="tns:CreateFolderPathType"/>
    <xs:complexType name="CreateFolderPathResponseType">
        <xs:complexContent mixed="false">
            <xs:extension base="tns:BaseResponseMessageType"/>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="CreateFolderPathResponse" type="tns:CreateFolderPathResponseType"/>
</xs:schema>
```
8 Appendix C: Product Behavior

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

- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Outlook 2013
- Microsoft Outlook 2016
- Microsoft Exchange Server 2019
- Microsoft Outlook 2019
- Microsoft Outlook 2021
- Microsoft Outlook 2024 Preview

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

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.
- A document revision that captures changes to protocol functionality.

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 None means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Revision class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix C: Product Behavior</td>
<td>Updated list of supported products.</td>
<td>Major</td>
</tr>
</tbody>
</table>
10 Index

A
Abstract data model
  server 12
Applicability 8
Archiving an item example 21
Attribute groups 11
Attributes 11

C
Capability negotiation 8
Change tracking 29
CLient
  overview 12
Complex types 11
Creating a folder path example 22

D
Data model - abstract
  server 12

E
Events
  local - server 20
    timer - server 20
Examples
  archiving an item 21
    creating a folder path 22
ExchangeServicePortType port type 12

F
Fields - vendor-extensible 9
Full WSDL 25
Full XML schema 27

G
Glossary 5
Groups 11

I
Implementer - security considerations 24
Index of security parameters 24
Informative references 7
Initialization
  server 12
  Introduction 5

L
Local events
  server 20

M
Message processing
  server 12
Messages
  attribute groups 11
  attributes 11
  complex types 11
  elements 10
  enumerated 10
  groups 11
  namespaces 10
  overview 10
  simple types 11
  syntax 10
  transport 10

N
Namespaces 10
Normative references 6

O
Operations
  ArchiveItem 12
    CreateFolderPath 16
  Overview (synopsis) 7

P
Parameters - security index 24
Port type
  ExchangeServicePortType 12
Preconditions 8
Prerequisites 8
Product behavior 28
Protocol Details
  overview 12

R
References 6
  informative 7
  normative 6
Relationship to other protocols 8

S
Security
  implementer considerations 24
  parameter index 24
Sequencing rules
  server 12
Server
  abstract data model 12
  ArchiveItem operation 12
  CreateFolderPath operation 16
  ExchangeServicePortType 12
  initialization 12
  local events 20
  message processing
    overview 12
sequencing rules 12
  timer events 20
  timers 12
Simple types 11
Standards assignments 9
Syntax
  messages - overview 10

T
Timer events
  server 20
Timers
    server 12
Tracking changes 29
Transport 10
Types
  complex 11
  simple 11

V
Vendor-extensible fields 9
Versioning 8

W
WSDL 25

X
XML schema 27