

[MS-OXWSCVTID]:

## Convert Item Identifier Web Service Protocol

---

### Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto: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. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

**Preliminary Documentation.** This Open Specification provides documentation for past and current releases and/or for the pre-release version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional

development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Preliminary

## Revision Summary

Date	Revision History	Revision Class	Comments
7/15/2009	1.0	Major	Initial Availability.
11/4/2009	1.1.0	Minor	Updated the technical content.
2/10/2010	1.2.0	Minor	Updated the technical content.
5/5/2010	1.2.1	Editorial	Revised and edited the technical content.
8/4/2010	2.0	Major	Significantly changed the technical content.
11/3/2010	2.1	Minor	Clarified the meaning of the technical content.
3/18/2011	3.0	Major	Significantly changed the technical content.
8/5/2011	3.1	Minor	Clarified the meaning of the technical content.
10/7/2011	4.0	Major	Significantly changed the technical content.
1/20/2012	5.0	Major	Significantly changed the technical content.
4/27/2012	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	5.1	Minor	Clarified the meaning of the technical content.
10/8/2012	6.0	Major	Significantly changed the technical content.
2/11/2013	6.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	6.0	No Change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	6.0	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	6.0	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	6.1	Minor	Clarified the meaning of the technical content.
7/31/2014	6.2	Minor	Clarified the meaning of the technical content.
10/30/2014	6.2	No Change	No changes to the meaning, language, or formatting of the technical content.
3/16/2015	7.0	Major	Significantly changed the technical content.
5/26/2015	7.0	No Change	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>6</b>
1.1	Glossary .....	6
1.2	References .....	7
1.2.1	Normative References .....	7
1.2.2	Informative References .....	8
1.3	Overview .....	8
1.4	Relationship to Other Protocols .....	8
1.5	Prerequisites/Preconditions .....	9
1.6	Applicability Statement .....	9
1.7	Versioning and Capability Negotiation .....	9
1.8	Vendor-Extensible Fields .....	9
1.9	Standards Assignments.....	9
<b>2</b>	<b>Messages.....</b>	<b>10</b>
2.1	Transport.....	10
2.2	Common Message Syntax .....	10
2.2.1	Namespaces .....	10
2.2.2	Messages.....	11
2.2.3	Elements .....	11
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
2.2.9	Common Data Structures .....	11
<b>3</b>	<b>Protocol Details .....</b>	<b>12</b>
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	ConvertId .....	12
3.1.4.1.1	Messages .....	13
3.1.4.1.1.1	tns:ConvertIdSoapIn Message .....	13
3.1.4.1.1.2	tns:ConvertIdSoapOut.....	14
3.1.4.1.2	Elements.....	14
3.1.4.1.2.1	m:ConvertId Element .....	15
3.1.4.1.2.2	m:ConvertIdResponse Element.....	15
3.1.4.1.3	Complex Types .....	15
3.1.4.1.3.1	m:ConvertIdResponseMessageType Complex Type.....	16
3.1.4.1.3.2	m:ConvertIdResponseType Complex Type .....	16
3.1.4.1.3.3	m:ConvertIdType Complex Type .....	16
3.1.4.1.3.4	t:AlternateIdBaseType Complex Type .....	17
3.1.4.1.3.5	t:AlternateIdType Complex Type.....	18
3.1.4.1.3.6	t:AlternatePublicFolderIdType Complex Type .....	18
3.1.4.1.3.7	t:AlternatePublicFolderItemIdType Complex Type .....	19
3.1.4.1.3.8	t:NonEmptyArrayOfAlternateIdsType Complex Type .....	19
3.1.4.1.4	Simple Types .....	20
3.1.4.1.4.1	t:IdFormatType Simple Type .....	20
3.1.4.1.5	Attributes .....	21
3.1.4.1.6	Groups.....	21
3.1.4.1.7	Attribute Groups.....	21
3.1.5	Timer Events.....	21

3.1.6	Other Local Events.....	22
<b>4</b>	<b>Protocol Examples.....</b>	<b>23</b>
<b>5</b>	<b>Security.....</b>	<b>25</b>
5.1	Security Considerations for Implementers.....	25
5.2	Index of Security Parameters.....	25
<b>6</b>	<b>Appendix A: Full WSDL.....</b>	<b>26</b>
<b>7</b>	<b>Appendix B: Full XML Schema.....</b>	<b>28</b>
7.1	Messages Schema.....	28
7.2	Types Schema.....	29
<b>8</b>	<b>Appendix C: Product Behavior.....</b>	<b>31</b>
<b>9</b>	<b>Change Tracking.....</b>	<b>32</b>
<b>10</b>	<b>Index.....</b>	<b>34</b>

Preliminary

# 1 Introduction

The Convert Item Identifier Web Service Protocol enables a client to convert identifier formats that can be used to locate items that are stored on the server.

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

## 1.1 Glossary

The following terms are specific to this document:

**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 over Secure Sockets Layer (HTTPS):** An extension of **HTTP** that securely encrypts and decrypts webpage requests.

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

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

**Simple Mail Transfer Protocol (SMTP):** A member of the TCP/IP suite of protocols that is used to transport Internet messages, as described in [\[RFC5321\]](#).

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

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

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

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

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

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

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

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

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

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

**XML:** The Extensible Markup Language, as described in [[XML1.0](#)].

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

**XML 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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[MS-OXPROPS] Microsoft Corporation, "[Exchange Server Protocols Master Property List](#)".

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

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

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

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

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

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

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

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

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

### 1.2.2 Informative References

[MS-OXDCLI] Microsoft Corporation, "[Autodiscover Publishing and Lookup Protocol](#)".

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

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

### 1.3 Overview

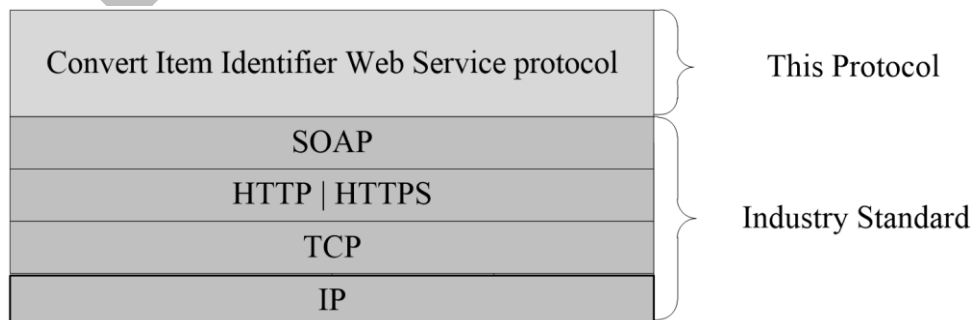
The Convert Item Identifier Web Service protocol enables clients to convert between alternative identifier formats for items that are stored by the server. Those identifiers can then be used to access the stored items by using other protocols and programmatic interfaces that are available on the server.

### 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 [\[MS-OXDCLI\]](#), to identify the target **endpoint** to use for each operation.

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

The Convert Item Identifier Web Service Protocol uses SOAP over **HTTP**, as described in [\[RFC2616\]](#), and SOAP over **HTTPS**, as described in [\[RFC2818\]](#), as shown in the following figure.





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

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

### 1.5 Prerequisites/Preconditions

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

### 1.6 Applicability Statement

The Convert Item Identifier Web Service protocol is applicable to clients that must obtain alternative identifiers to an item that is stored on the server, and then use those identifiers to utilize other protocols or application programming interfaces to access the stored item.

### 1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses SOAP 1.1, as specified in section [2.1](#).
- **Protocol Versions:** This protocol specifies only one **WSDL port type** version. The **WSDL** version of the request is identified by using the **RequestServerVersion** element, as described in [\[MS-OXWSCDATA\]](#) section 2.2.3.11, and the version of the server responding to the request is identified by using the **ServerVersionInfo** element, as described in [\[MS-OXWSCDATA\]](#) section 2.2.3.12.
- **Security and Authentication Methods:** This protocol relies on the web server that is hosting it to perform authentication.
- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section [3.1.4](#).
- **Capability Negotiation:** None.

### 1.8 Vendor-Extensible Fields

None.

### 1.9 Standards Assignments

None.

## 2 Messages

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

### 2.1 Transport

The **SOAP** version supported is SOAP 1.1. For details, see [\[SOAP1.1\]](#).

This protocol relies on the web server that hosts the application to perform authentication. This protocol **MUST** support SOAP over HTTP, as specified in [\[RFC2616\]](#). The protocol **SHOULD** use secure communications via HTTPS, as defined in [\[RFC2818\]](#).

### 2.2 Common Message Syntax

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

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

#### 2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
<b>soap</b>	<a href="http://schemas.xmlsoap.org/wsdl/soap/">http://schemas.xmlsoap.org/wsdl/soap/</a>	<a href="#">[SOAP1.1]</a>
<b>tns</b>	<a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a>	
<b>xs</b>	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	<a href="#">[XMLSCHEMA1]</a> <a href="#">[XMLSCHEMA2]</a>
<b>(none)</b>	<a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a>	
<b>wsdl</b>	<a href="http://schemas.xmlsoap.org/wsdl/">http://schemas.xmlsoap.org/wsdl/</a>	<a href="#">[WSDL]</a>
<b>t</b>	<a href="http://schemas.microsoft.com/exchange/services/2006/types">http://schemas.microsoft.com/exchange/services/2006/types</a>	
<b>m</b>	<a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a>	

### **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.

### **2.2.9 Common Data Structures**

This specification does not define any common XML schema data structures.

### 3 Protocol Details

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

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

#### 3.1 ExchangeServicePortType Server Details

##### 3.1.1 Abstract Data Model

The Convert Item Identifier Web Service Protocol is a stateless protocol.

##### 3.1.2 Timers

None.

##### 3.1.3 Initialization

None.

##### 3.1.4 Message Processing Events and Sequencing Rules

This protocol includes the operation that is listed in the following table.

Operation name	Description
ConvertId	Converts the supplied item identifier into a different format.

###### 3.1.4.1 ConvertId

The **ConvertId** operation SHOULD [<1>](#) convert the item identifier supplied by the client into a different identifier format.

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

```
<wsdl:operation name="ConvertId">
  <wsdl:input message="tns:ConvertIdSoapIn"/>
  <wsdl:output message="tns:ConvertIdSoapOut"/>
</wsdl:operation>
```

The following is the WSDL binding specification of the operation.

```
<wsdl:operation name="ConvertId">
  <soap:operation
    soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/ConvertId" />
  <wsdl:input>
    <soap:header message="tns:ConvertIdSoapIn" part="Impersonation" use="literal"/>
    <soap:header message="tns:ConvertIdSoapIn" part="RequestVersion" use="literal"/>
    <soap:body parts="request" use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="ConvertIdResult" use="literal" />
    <soap:header message="tns:ConvertIdSoapOut" part="ServerVersion" use="literal"/>
  </wsdl:output>
</wsdl:operation>
```

Servers that implement this protocol SHOULD [<2>](#) give clients the option to implement the **Impersonation SOAP header**.

### 3.1.4.1.1 Messages

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

Message name	Description
<b>ConvertIdSoapIn</b>	Specifies the <b>SOAP message</b> that requests the conversion of one or more identifiers.
<b>ConvertIdSoapOut</b>	Specifies the SOAP message that is returned by the server in response to a request to convert identifiers.

#### 3.1.4.1.1.1 tns:ConvertIdSoapIn Message

The **ConvertIdSoapIn** WSDL message specifies the SOAP message that requests the conversion of one or more identifiers.

The following is the WSDL message specification of the **ConvertIdSoapIn** message.

```
<wsdl:message name="ConvertIdSoapIn">
  <wsdl:part name="request" element="tns:ConvertId"/>
  <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
  <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
```

Servers that implement this protocol SHOULD [<3>](#) give clients the option to implement the **Impersonation** part.

The **ConvertIdSoapIn** WSDL message is the input message for the **SOAP action** <http://schemas.microsoft.com/exchange/services/2006/messages/ConvertId>.

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

Part name	Element/type	Description
<b>request</b>	<b>tns:ConvertId</b> (section <a href="#">3.1.4.1.2.1</a> )	Specifies the <b>SOAP body</b> of the request to convert identifiers.
<b>Impersonation</b>	<b>t:ExchangeImpersonation</b> ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating. This part is optional.
<b>RequestVersion</b>	<b>t:RequestServerVersion</b> ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the <b>ConvertId</b> operation request.

### 3.1.4.1.1.2 tns:ConvertIdSoapOut

The **ConvertIdSoapOut** WSDL message specifies the SOAP message that is returned in response to a request to convert one or more identifiers

The following is the WSDL message specification of the **ConvertIdSoapOut** message.

```
<wsdl:message name="ConvertIdSoapOut">
  <wsdl:part name="ConvertIdResult" element="tns:ConvertIdResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

The **ConvertIdSoapOut** WSDL message is the output message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/ConvertId>.

The parts of the **ConvertIdSoapOut** WSDL message are specified in the following table.

Part name	Element/type	Description
<b>ConvertIdResult</b>	<b>tns:ConvertIdResponse</b> (section <a href="#">3.1.4.1.2.2</a> )	Specifies the SOAP body of the response to a <b>ConvertId</b> request.
<b>ServerVersion</b>	<b>t:ServerVersionInfo</b> ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

A successful **ConvertId** WSDL operation request returns a **ConvertIdResponseMessage** element with the **ResponseClass** (as specified in [MS-OXWSCDATA] section 2.2.4.65) attribute set to "Success". The **ResponseCode** (as specified in [MS-OXWSCDATA] section 2.2.4.65) element of the **ConvertIdResponseMessage** element is set to "NoError".

If the **ConvertId** WSDL operation is not successful, it returns a **ConvertIdResponseMessage** element with the **ResponseClass** (as specified in [MS-OXWSCDATA] section 2.2.4.65) attribute set to "Error". The **ResponseCode** (as specified in [MS-OXWSCDATA] section 2.2.4.65) element of the **ConvertIdResponseMessage** element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

### 3.1.4.1.2 Elements

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

Element name	Description
--------------	-------------

Element name	Description
<b>ConvertId</b>	Specifies a request to convert a supplied list of source item identifiers into the requested destination format.
<b>ConvertIdResponse</b>	Specifies a response from the server, and contains the alternate item identifiers for the supplied items in the requested format.

### 3.1.4.1.2.1 m:ConvertId Element

The **ConvertId** element defines a request to convert a supplied list of source item identifiers into the requested destination format. The **ConvertId** element is of type **ConvertIdType**, as specified in section [3.1.4.1.3.3](#).

```
<xs:element name="ConvertId"
  type="m:ConvertIdType"
 />
```

### 3.1.4.1.2.2 m:ConvertIdResponse Element

The **ConvertIdResponse** element specifies a response from the server, and contains the alternate item identifiers for the supplied items in the requested format. The **ConvertIdResponse** element is of type **ConvertIdResponseType**, as specified in section [3.1.4.1.3.2](#).

```
<xs:element name="ConvertIdResponse"
  type="m:ConvertIdResponseType"
 />
```

### 3.1.4.1.3 Complex Types

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

Complex type name	Description
<b>ConvertIdResponseMessageType</b>	Specifies the status and result of a <b>ConvertId</b> operation request.
<b>ConvertIdResponseType</b>	Specifies the contents of a response to a <b>ConvertId</b> operation request.
<b>ConvertIdType</b>	Specifies the contents of a <b>ConvertId</b> operation request.
<b>AlternateIdBaseType</b>	Specifies the base container type for the attributes that specify information about the returned item identifier.
<b>AlternateIdType</b>	Specifies the information that is provided with a returned <b>mailbox</b> folder or mailbox item identifier.
<b>AlternatePublicFolderIdType</b>	Specifies the information that is provided with a returned <b>public folder</b> item identifier.
<b>AlternatePublicFolderItemIdType</b>	Specifies the information that is provided with a returned item that is located in a public folder.

Complex type name	Description
<b>NonEmptyArrayOfAlternateIdsType</b>	Specifies a container for one or more item identifiers.

### 3.1.4.1.3.1 m:ConvertIdResponseMessageType Complex Type

The **ConvertIdResponseMessageType** complex type specifies the status and result of a **ConvertId** operation request. The **ConvertIdResponseMessageType** complex type extends the **ResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.65.

```
<xs:complexType name="ConvertIdResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="AlternateId"
          type="t:AlternateIdBaseType"
          maxOccurs="1"
          minOccurs="0"
        />
      </xs:sequence>
    </xs:complexContent>
  </xs:complexType>
```

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

Element name	Type	Description
<b>AlternateId</b>	<b>t:AlternateIdBaseType</b> (section <a href="#">3.1.4.1.3.4</a> )	Specifies a converted identifier in the response.

### 3.1.4.1.3.2 m:ConvertIdResponseType Complex Type

The **ConvertIdResponseType** complex type specifies the contents of a response to a **ConvertId** operation request. The **ConvertIdResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.18.

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

### 3.1.4.1.3.3 m:ConvertIdType Complex Type



The **ConvertIdType** complex type specifies the contents of a **ConvertId** operation request. The **ConvertIdType** complex type extends the **BaseRequestType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.17.

```
<xs:complexType name="ConvertIdType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="SourceIds"
          type="t:NonEmptyArrayOfAlternateIdsType"
        />
      </xs:sequence>
      <xs:attribute name="DestinationFormat"
        type="t:IdFormatType"
        use="required"
      />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
<b>SourceIds</b>	<b>t:NonEmptyArrayOfAlternateIdsType</b> (section <a href="#">3.1.4.1.3.8</a> ).	Specifies the source identifiers to convert.

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

Attribute name	Type	Description
<b>DestinationFormat</b>	<b>t:IdFormatType</b> (section <a href="#">3.1.4.1.4.1</a> ).	Specifies the identifier format that will be returned for all the converted identifiers.

#### 3.1.4.1.3.4 t:AlternateIdBaseType Complex Type

The **AlternateIdBaseType** complex type specifies the base container type for the attributes that specify information about the returned item identifier.

```
<xs:complexType name="AlternateIdBaseType"
  abstract="true"
>
  <xs:attribute name="Format"
    type="t:IdFormatType"
    use="required"
  />
</xs:complexType>
```

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

Attribute name	Type	Description
<b>Format</b>	<b>t:IdFormatType</b> (section <a href="#">3.1.4.1.4.1</a> ).	Specifies the identifier format.

### 3.1.4.1.3.5 t:AlternateIdType Complex Type

The **AlternateIdType** complex type specifies the information that is provided with a returned mailbox folder or mailbox item identifier. The **AlternateIdType** complex type extends the **AlternateIdBaseType** complex type, as specified in section [3.1.4.1.3.4](#).

```
<xs:complexType name="AlternateIdType">
  <xs:complexContent>
    <xs:extension
      base="t:AlternateIdBaseType"
    >
      <xs:attribute name="Id"
        type="xs:string"
        use="required"
      />
      <xs:attribute name="Mailbox"
        type="t:NonEmptyStringType"
        use="required"
      />
      <xs:attribute name="IsArchive"
        type="xs:boolean"
      />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Attribute name	Type	Description
<b>Id</b>	<b>xs:string</b> <a href="#">[XMLSCHEMA2]</a>	Specifies the source identifier in a request and the destination identifier in a response. This attribute <b>MUST</b> be present. The maximum length is 512 bytes after base64 decoding.
<b>Mailbox</b>	<b>t:NonEmptyStringType</b> ( <a href="#">[MS-OXWSCDATA]</a> section 2.2.5.20)	Specifies the mailbox primary <b>SMTP</b> address of the identifier to convert. This attribute <b>MUST</b> be present.
<b>IsArchive</b>	<b>xs:boolean</b> <a href="#">[XMLSCHEMA2]</a>	Specifies whether the identifier represents an archived item. This attribute <b>MUST</b> be present if the item is an archived item. Otherwise, this attribute is optional. <a href="#">&lt;4&gt;</a> If this attribute is not present, the value is assumed to equal "false".

### 3.1.4.1.3.6 t:AlternatePublicFolderIdType Complex Type

The **AlternatePublicFolderIdType** complex type specifies the information that is provided with a returned public folder item identifier. The **AlternatePublicFolderIdType** complex type extends the **AlternateIdBaseType** complex type, as specified in section [3.1.4.1.3.4](#).

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

```

<xs:complexContent>
  <xs:extension
    base="t:AlternateIdBaseType"
  >
    <xs:attribute name="FolderId"
      type="xs:string"
      use="required"
    />
  </xs:extension>
</xs:complexContent>
</xs:complexType>

```

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

Attribute name	Type	Description
<b>FolderId</b>	<b>xs:string</b> <a href="#">[XMLSCHEMA2]</a>	Specifies the identifier of a folder to convert. The maximum length is 512 bytes after base64 decoding.

#### 3.1.4.1.3.7 t:AlternatePublicFolderItemIdType Complex Type

The **AlternatePublicFolderItemIdType** complex type specifies the information that is provided with a returned item that is located in a public folder. The **AlternatePublicFolderItemIdType** complex type extends the **AlternatePublicFolderIdType** complex type, as specified in section [3.1.4.1.3.6](#).

```

<xs:complexType name="AlternatePublicFolderItemIdType">
  <xs:complexContent>
    <xs:extension
      base="t:AlternatePublicFolderIdType"
    >
      <xs:attribute name="ItemId"
        type="xs:string"
        use="required"
      />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

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

Attribute name	Type	Description
<b>ItemId</b>	<b>xs:string</b> <a href="#">[XMLSCHEMA2]</a>	Specifies the public folder item identifier to convert. The maximum length is 512 bytes after base64 decoding.

#### 3.1.4.1.3.8 t:NonEmptyArrayOfAlternateIdsType Complex Type

The **NonEmptyArrayOfAlternateIdsType** complex type specifies a container for one or more item identifiers. Each individual identifier MUST be an **AlternateIdType** complex type, as specified in section [3.1.4.1.3.5](#), an **AlternatePublicFolderIdType** complex type, as specified in section

[3.1.4.1.3.6](#), or an **AlternatePublicFolderItemIdType** complex type, as specified in section [3.1.4.1.3.7](#). The **NonEmptyArrayOfAlternateIdsType** complex type does not extend any other complex type.

```
<xs:complexType name="NonEmptyArrayOfAlternateIdsType">
  <xs:choice
    maxOccurs="unbounded"
  >
    <xs:element name="AlternateId"
      type="t:AlternateIdType"
    />
    <xs:element name="AlternatePublicFolderId"
      type="t:AlternatePublicFolderIdType"
    />
    <xs:element name="AlternatePublicFolderItemId"
      type="t:AlternatePublicFolderItemIdType"
    />
  </xs:choice>
</xs:complexType>
```

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

Element name	Type	Description
<b>AlternateId</b>	<b>t:AlternateIdType</b> (section 3.1.4.1.3.5)	Specifies an item or folder identifier to convert.
<b>AlternatePublicFolderId</b>	<b>t:AlternatePublicFolderIdType</b> (section 3.1.4.1.3.6)	Specifies a public folder identifier to convert.
<b>AlternatePublicFolderItemId</b>	<b>t:AlternatePublicFolderItemIdType</b> (section 3.1.4.1.3.7)	Specifies a public folder item identifier to convert.

### 3.1.4.1.4 Simple Types

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

Simple type name	Description
<b>IdFormatType</b>	Specifies the item identifier format in both the client request and for each returned identifier.

#### 3.1.4.1.4.1 t:IdFormatType Simple Type

The **IdFormatType** simple type specifies the item identifier format in both the client request and for each returned identifier.

```
<xs:simpleType name="IdFormatType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="EwsLegacyId"
    />
  </xs:restriction>
</xs:simpleType>
```

```

/>
<xs:enumeration
  value="EwsId"
/>
<xs:enumeration
  value="EntryId"
/>
<xs:enumeration
  value="HexEntryId"
/>
<xs:enumeration
  value="StoreId"
/>
<xs:enumeration
  value="OwaId"
/>
</xs:restriction>
</xs:simpleType>

```

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

Value	Meaning
<b>EwsLegacyId</b>	Specifies that the identifier is in the format that is used by the legacy version of the protocol. MAY<5> be present.
<b>EwsId</b>	Specifies that the identifier is in the format that is used by the current version of the protocol.
<b>EntryId</b>	Specifies that the identifier is in the format that can be used with the <b>PidTagEntryId</b> property, as specified in <a href="#">[MS-OXPROPS]</a> section 2.674.
<b>HexEntryId</b>	Specifies that the identifier is a hexadecimal-encoded representation of the format that can be used with the <b>PidTagEntryId</b> property.
<b>StoreId</b>	Specifies that the identifier is in a format that is recognized only by the server.
<b>OwaId</b>	Specifies that the identifier is in a format that is used by the web-based client capability of the server, if one exists.

#### 3.1.4.1.5 Attributes

None.

#### 3.1.4.1.6 Groups

None.

#### 3.1.4.1.7 Attribute Groups

None.

### 3.1.5 Timer Events

None.

### 3.1.6 Other Local Events

None.

Preliminary

## 4 Protocol Examples

In this example, the protocol client constructs the following SOAP message to request the conversion of an **OwaId** to an **EwsId**.

```
<?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_SP1" />

  </soap:Header>

  <soap:Body>

    <m:ConvertId DestinationFormat="EwsId">

      <m:SourceIds>

        <t:AlternateId Format="OwaId"
Id="RgAAAAAzFgcvl%2fwTo7NTrcPscM3BwDHxFWht3DVTTrtUQ
kRbTXDCAAAAMUgfAADHxFWht3DVTTrtUQkRbTXDCAAAAMW98AAAJ" Mailbox="user1@example.com" />

      </m:SourceIds>

    </m:ConvertId>

  </soap:Body>

</soap:Envelope>
```

The protocol server responds with the converted identifier.

```
<?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="218"
MinorBuildNumber="12" 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:ConvertIdResponse
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
```

```
<m:ResponseMessages>
  <m:ConvertIdResponseMessage ResponseClass="Success">
    <m:ResponseCode>NoError</m:ResponseCode>
    <m:AlternateId xsi:type="t:AlternateIdType" Format="EwsId"
  Id="AAMkAGE4NTY1YWNjLT
  JkNmMtNGIwYy1hZWFlLTcwNjYzMDNjYjlkZQBGAIAAAAzFgecVl/WTo7NTrcPscM3
  BwDHxFWht3DVTrtUQkRbTXDCAAAMUgfaADHxFWht3DVTrtUQkRbTXDCAAAMW98AAA="
  Mailbox="User1@example.com" />
  </m:ConvertIdResponseMessage>
</m:ResponseMessages>
</m:ConvertIdResponse>
</s:Body>
</s:Envelope>
```

Preliminary



## 5 Security

### 5.1 Security Considerations for Implementers

The Convert Item Identifier Web Service Protocol does not use any additional security mechanisms.

### 5.2 Index of Security Parameters

None.

Preliminary

## 6 Appendix A: Full WSDL

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

File name	Description	Section
MS-OXWSCVTID.wsdl	Contains the WSDL for the implementation of this protocol.	6
MS-OXWSCVTID-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	<a href="#">7.1</a>
MS-OXWSCVTID-types.xsd	Contains the XML schema type definitions that are used in this protocol.	<a href="#">7.2</a>

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-OXWSCVTID-types.xsd or MS-OXWSCVTID-messages.xsd schemas have to be placed in the common folder along with the files.

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

```
<?xml version="1.0" encoding="utf-8"?>
<wSDL:definitions xmlns:soap="http://schemas.xmlsoap.org/wSDL/soap/"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
  <wSDL:types>
    <xs:schema id="messages" elementFormDefault="qualified" version="Exchange2016"
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
      <xs:include schemaLocation="MS-OXWSCVTID-messages.xsd"/>
    </xs:schema>
    <xs:schema id="types" elementFormDefault="qualified" version="Exchange2016"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:include schemaLocation="MS-OXWSCVTID-types.xsd"/>
    </xs:schema>
  </wSDL:types>
  <wSDL:message name="ConvertIdSoapIn" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/">
    <wSDL:part name="request" element="tns:ConvertId"/>
    <wSDL:part name="Impersonation" element="t:ExchangeImpersonation"/>
    <wSDL:part name="RequestVersion" element="t:RequestServerVersion"/>
  </wSDL:message>
  <wSDL:message name="ConvertIdSoapOut" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/">
    <wSDL:part name="ConvertIdResult" element="tns:ConvertIdResponse"/>
    <wSDL:part name="ServerVersion" element="t:ServerVersionInfo"/>
  </wSDL:message>
  <wSDL:portType name="ExchangeServicePortType">
    <wSDL:operation name="ConvertId" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/">
      <wSDL:input message="tns:ConvertIdSoapIn"/>
      <wSDL:output message="tns:ConvertIdSoapOut"/>
    </wSDL:operation>
  </wSDL:portType>
  <wSDL:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
    <wSDL:documentation xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/">
      <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0" xmlns:wsi="http://ws-
i.org/schemas/conformanceClaim"/>
    </wSDL:documentation>
  </wSDL:binding>
</wSDL:definitions>
```

```
</wsdl:documentation>
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <wsdl:operation name="ConvertId" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/ConvertId"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
      <wsdl:input>
        <soap:header message="tns:ConvertIdSoapIn" part="RequestVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
          <soap:header message="tns:ConvertIdSoapIn" part="Impersonation" use="literal"/>
          <soap:body parts="request" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
        </wsdl:input>
        <wsdl:output>
          <soap:body parts="ConvertIdResult" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
          <soap:header message="tns:ConvertIdSoapOut" part="ServerVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  </wsdl:definitions>
```

Preliminary

## 7 Appendix B: Full XML Schema

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

Schema name	Prefix	Section
Messages schema	m:	<a href="#">7.1</a>
Types schema	t:	<a href="#">7.2</a>

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

### 7.1 Messages Schema

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

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

File name	Defining specification/section
MS-OXWSCDATA-messages.xsd	<a href="#">[MS-OXWSCDATA]</a> section 7.1
MS-OXWSCVTID-types.xsd	<a href="#">7.2</a>

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
  elementFormDefault="qualified" version="Exchange2016" id="messages">
  <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
    schemaLocation="MS-OXWSCVTID-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
  <xs:complexType name="ConvertIdResponseMessageType"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:complexContent>
      <xs:extension base="m:ResponseMessageType">
        <xs:sequence>
          <xs:element name="AlternateId" type="t:AlternateIdBaseType" minOccurs="0"
            maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="ConvertIdResponseType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="ConvertIdType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:element name="SourceIds" type="t:NonEmptyArrayOfAlternateIdsType" minOccurs="1"
            maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

```

        </xs:sequence>
        <xs:attribute name="DestinationFormat" type="t:IdFormatType" use="required"/>
    </xs:extension>
</xs:complexType>
</xs:complexType>
<xs:element name="ConvertId" type="m:ConvertIdType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
<xs:element name="ConvertIdResponse" type="m:ConvertIdResponseType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
</xs:schema>

```

## 7.2 Types Schema

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

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

File name	Defining specification/section
MS-OXWSCDATA-types.xsd	<a href="#">[MS-OXWSCDATA]</a> section 7.2

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
elementFormDefault="qualified"
version="Exchange2016"
id="types">
<xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
<xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
<xs:complexType name="AlternateIdBaseType" abstract="true"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:attribute name="Format" type="t:IdFormatType" use="required"/>
</xs:complexType>
<xs:complexType name="AlternateIdType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:complexContent>
<xs:extension base="t:AlternateIdBaseType">
<xs:attribute name="Id" type="xs:string" use="required"/>
<xs:attribute name="Mailbox" type="t:NonEmptyStringType" use="required"/>
<xs:attribute name="IsArchive" type="xs:boolean" use="optional"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="AlternatePublicFolderIdType"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:complexContent>
<xs:extension base="t:AlternateIdBaseType">
<xs:attribute name="FolderId" type="xs:string" use="required"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="AlternatePublicFolderItemIdType"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:complexContent>
<xs:extension base="t:AlternatePublicFolderIdType">
<xs:attribute name="ItemId" type="xs:string" use="required"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

```
<xs:simpleType name="IdFormatType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EwsLegacyId"/>
    <xs:enumeration value="EwsId"/>
    <xs:enumeration value="EntryId"/>
    <xs:enumeration value="HexEntryId"/>
    <xs:enumeration value="StoreId"/>
    <xs:enumeration value="OwaId"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="NonEmptyArrayOfAlternateIdsType"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:choice minOccurs="1" maxOccurs="unbounded">
    <xs:element name="AlternateId" type="t:AlternateIdType"/>
    <xs:element name="AlternatePublicFolderId" type="t:AlternatePublicFolderIdType"/>
    <xs:element name="AlternatePublicFolderItemId"
type="t:AlternatePublicFolderItemIdType"/>
  </xs:choice>
</xs:complexType>
</xs:schema>
```

Preliminary

## 8 Appendix C: Product Behavior

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

- Microsoft Exchange Server 2007 Service Pack 1 (SP1)
- Microsoft Exchange Server 2007 Service Pack 2 (SP2)
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016 Preview
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016 Preview

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

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

<1> [Section 3.1.4.1](#): Microsoft Outlook 2010, Outlook 2013, and Outlook 2016 Preview use the **ConvertId** operation to convert an "EntryId" value to an "EwsId" value.

<2> [Section 3.1.4.1](#): Exchange 2007 SP1, Exchange 2010, and Microsoft Exchange Server 2010 Service Pack 1 (SP1) do not implement the **Impersonation** SOAP header. The **Impersonation** header was introduced in Microsoft Exchange Server 2010 Service Pack 2 (SP2).

<3> [Section 3.1.4.1.1.1](#): Exchange 2007 SP1, Exchange 2010, and Exchange 2010 SP1 do not implement the **Impersonation** header. The **Impersonation** header was introduced in Exchange 2010 SP2.

<4> [Section 3.1.4.1.3.5](#): Exchange 2007 SP1 and Exchange 2010 do not include the **IsArchive** attribute.

<5> [Section 3.1.4.1.4.1](#): The "EwsLegacyId" value is applicable for identifiers created by Microsoft Exchange Server 2007.

## 9 Change Tracking

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

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

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

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

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

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

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

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

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

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



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

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

The changes made to this document are listed in the following table. For more information, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">8</a> Appendix C: Product Behavior	Added Exchange 2016 and Outlook 2016 to list of applicable products.	Y	Content update.

Preliminary

## 10 Index

### A

Abstract data model  
[server](#) 12  
[Applicability](#) 9  
[Attribute groups](#) 11  
[Attributes](#) 11

### C

[Capability negotiation](#) 9  
[Change tracking](#) 32  
[Common data structures](#) 11  
[Complex types](#) 11

### D

Data model - abstract  
[server](#) 12

### E

Events  
[local - server](#) 22  
[timer - server](#) 21  
Examples  
[overview](#) 23

### F

[Fields - vendor-extensible](#) 9  
[Full WSDL](#) 26  
[Full XML schema](#) 28  
[Messages Schema](#) 28  
[Types Schema](#) 29

### G

[Glossary](#) 6  
[Groups](#) 11

### I

[Implementer - security considerations](#) 25  
[Index of security parameters](#) 25  
[Informative references](#) 8  
Initialization  
[server](#) 12  
[Introduction](#) 6

### L

Local events  
[server](#) 22

### M

Message processing  
[server](#) 12

### Messages

[attribute groups](#) 11  
[attributes](#) 11  
[common data structures](#) 11  
[complex types](#) 11  
[elements](#) 11  
[enumerated](#) 11  
[groups](#) 11  
[namespaces](#) 10  
[simple types](#) 11  
[syntax](#) 10  
[transport](#) 10

### N

[Namespaces](#) 10  
[Normative references](#) 7

### O

Operations  
[ConvertId](#) 12  
[Overview \(synopsis\)](#) 8

### P

[Parameters - security index](#) 25  
[Preconditions](#) 9  
[Prerequisites](#) 9  
[Product behavior](#) 31  
Protocol Details  
[overview](#) 12

### R

[References](#) 7  
[informative](#) 8  
[normative](#) 7  
[Relationship to other protocols](#) 8

### S

Security  
[implementer considerations](#) 25  
[parameter index](#) 25  
Sequencing rules  
[server](#) 12  
Server  
[abstract data model](#) 12  
[ConvertId operation](#) 12  
[initialization](#) 12  
[local events](#) 22  
[message processing](#) 12  
[sequencing rules](#) 12  
[timer events](#) 21  
[timers](#) 12  
[Simple types](#) 11  
[Standards assignments](#) 9  
Syntax  
[messages - overview](#) 10

## T

Timer events

[server](#) 21

Timers

[server](#) 12

[Tracking changes](#) 32

[Transport](#) 10

Types

[complex](#) 11

[simple](#) 11

## V

[Vendor-extensible fields](#) 9

[Versioning](#) 9

## W

[WSDL](#) 26

## X

[XML schema](#) 28

[Messages Schema](#) 28

[Types Schema](#) 29

Preliminary