

[MS-OXWSCVTID]: Convert Item Identifier Web Service Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) 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 (beta) 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.

Revision Summary

Date	Revision History	Revision Class	Comments
07/15/2009	1.0	Major	Initial Availability.
11/04/2009	1.1.0	Minor	Updated the technical content.
02/10/2010	1.2.0	Minor	Updated the technical content.
05/05/2010	1.2.1	Editorial	Revised and edited the technical content.
08/04/2010	2.0	Major	Significantly changed the technical content.
11/03/2010	2.1	Minor	Clarified the meaning of the technical content.
03/18/2011	3.0	Major	Significantly changed the technical content.
08/05/2011	3.1	Minor	Clarified the meaning of the technical content.
10/07/2011	4.0	Major	Significantly changed the technical content.
01/20/2012	5.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	5
1.2.1 Normative References	5
1.2.2 Informative References	6
1.3 Overview	6
1.4 Relationship to Other Protocols	6
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	7
1.8 Vendor-Extensible Fields	7
1.9 Standards Assignments	8
2 Messages	9
2.1 Transport	9
2.2 Common Message Syntax	9
2.2.1 Namespaces	9
2.2.2 Messages	9
2.2.3 Elements	10
2.2.4 Complex Types	10
2.2.5 Simple Types	10
2.2.6 Attributes	10
2.2.7 Groups	10
2.2.8 Attribute Groups	10
2.2.9 Common Data Structures	10
3 Protocol Details	11
3.1 ExchangeServicePortType Server Details	11
3.1.1 Server Abstract Data Model	11
3.1.2 Server Timers	11
3.1.3 Server Initialization	11
3.1.4 Server Message Processing Events and Sequencing	11
3.1.4.1 ConvertId	11
3.1.4.1.1 ConvertID Messages	12
3.1.4.1.1.1 tns:ConvertIdSoapIn Message	12
3.1.4.1.1.2 tns:ConvertIdSoapOut	13
3.1.4.1.2 ConvertId Elements	13
3.1.4.1.2.1 m:ConvertId Element	13
3.1.4.1.2.2 m:ConvertIdResponse Element	14
3.1.4.1.3 ConvertId Complex Types	14
3.1.4.1.3.1 m:ConvertIdResponseMessageType Complex Type	14
3.1.4.1.3.2 m:ConvertIdResponseType Complex Type	15
3.1.4.1.3.3 m:ConvertIdType Complex Type	15
3.1.4.1.3.4 t:AlternateIdBaseType Complex Type	16
3.1.4.1.3.5 t:AlternateIdType Complex Type	16
3.1.4.1.3.6 t:AlternatePublicFolderIdType Complex Type	17
3.1.4.1.3.7 t:AlternatePublicFolderItemIdType Complex Type	18
3.1.4.1.3.8 t:NonEmptyArrayOfAlternateIdsType Complex Type	18
3.1.4.1.4 ConvertId Simple Types	19
3.1.4.1.4.1 t:IdFormatType Simple Type	19

3.1.5 Server Timer Events	20
3.1.6 Server Other Local Events	20
4 Protocol Examples	21
5 Security	22
5.1 Security Considerations for Implementers	22
5.2 Index of Security Parameters	22
6 Appendix A: Full WSDL	23
7 Appendix B: Full XML Schema	25
7.1 Messages Schema	25
7.2 Types Schema	26
8 Appendix C: Product Behavior	28
9 Change Tracking	29
10 Index	31

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 contain RFC 2119 language. Sections 1.5 and 1.9 are also normative but cannot contain RFC 2119 language. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
XML

The following terms are defined in [\[MS-OXGLOS\]](#):

endpoint
mailbox
public folder
Simple Mail Transfer Protocol (SMTP)
Simple Object Access Protocol (SOAP)
SOAP action
SOAP body
SOAP header
SOAP message
Uniform Resource Locator (URL)
Web Services Description Language (WSDL)
WSDL message
WSDL port type
XML namespace
XML schema

The following terms are specific to this document:

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

1.2 References

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

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

[MS-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>

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <http://www.ietf.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.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

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

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

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

[MS-OXGLOS] Microsoft Corporation, "[Exchange Server Protocols Master Glossary](#)".

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

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

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 (4)** 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.

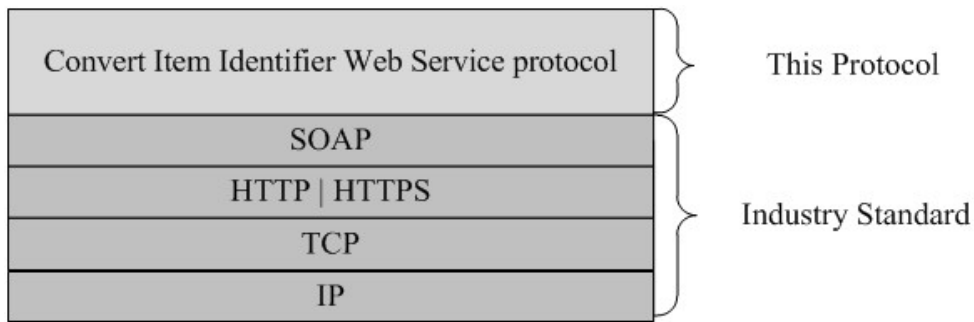


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

The endpoint(4) **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 (4) 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 2.2.4.9, and the version of the server responding to the request is identified by using the **ServerVersionInfo** element, as described in [\[MS-OXWSCDATA\]](#) section 2.2.4.10.
- **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.

Preliminary

2 Messages

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

2.1 Transport

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

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

2.2 Common Message Syntax

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

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
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/exchange/services/2006/messages	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
(none)	http://schemas.microsoft.com/exchange/services/2006/messages	
wSDL	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	

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 Server Abstract Data Model

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

3.1.2 Server Timers

None.

3.1.3 Server Initialization

None.

3.1.4 Server Message Processing Events and Sequencing

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 ConvertID Messages

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

Message name	Description
ConvertIdSoapIn	Specifies the SOAP message that requests the conversion of one or more identifiers.
ConvertIdSoapOut	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
request	tns:ConvertId (section 3.1.4.1)	Specifies the SOAP body of the request to convert identifiers.
Impersonation	t:ExchangeImpersonation ([MS-OXWSCDATA] section 2.2.4.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating. This part is optional.
RequestVersion	t:RequestServerVersion ([MS-OXWSCDATA] section 2.2.4.9)	Specifies a SOAP header that identifies the schema version for the ConvertId 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
ConvertIdResult	tns:ConvertIdResponse (section 3.1.4.1.2.2)	Specifies the SOAP body of the response to a ConvertId request.
ServerVersion	t:ServerVersionInfo ([MS-OXWSCDATA] section 2.2.4.10)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.1.2 ConvertId Elements

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

Element name	Description
ConvertId	Specifies a request to convert a supplied list of source item identifiers into the requested destination format.
ConvertIdResponse	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 ConvertId Complex Types

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

Complex type name	Description
ConvertIdResponseMessageType	Specifies the status and result of a ConvertId operation request.
ConvertIdResponseType	Specifies the contents of a response to a ConvertId operation request.
ConvertIdType	Specifies the contents of a ConvertId operation request.
AlternateIdBaseType	Specifies the base container type for the attributes that specify information about the returned item identifier.
AlternateIdType	Specifies the information that is provided with a returned mailbox folder or mailbox item identifier.
AlternatePublicFolderIdType	Specifies the information that is provided with a returned public folder item identifier.
AlternatePublicFolderItemIdType	Specifies the information that is provided with a returned item that is located in a public folder.
NonEmptyArrayOfAlternateIdsType	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 2.2.3.57.

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

```

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

Element name	Type	Description
AlternateId	t:AlternateIdBaseType (section 3.1.4.1.3.4)	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 2.2.3.16.

```

<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 2.2.3.15.

```

<xs:complexType name="ConvertIdType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    />
  </xs:complexContent>
  <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
SourceIds	t:NonEmptyArrayOfAlternateIdsType (section 3.1.4.1.3.8).	Specifies the source identifiers to convert.

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

Attribute name	Type	Description
DestinationFormat	t:IdFormatType (section 3.1.4.1.4.1).	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
Format	t:IdFormatType (section 3.1.4.1.4.1).	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
Id	xs:string [XMLSCHEMA2]	Specifies the source identifier in a request and the destination identifier in a response. This attribute MUST be present. The maximum length is 512 bytes after base64 decoding.
Mailbox	t:NonEmptyStringType ([MS-OXWSCDATA] section 2.2.2.19)	Specifies the mailbox primary SMTP address of the identifier to convert. This attribute MUST be present.
IsArchive	xs:boolean [XMLSCHEMA2]	Specifies whether the identifier represents an archived item. This attribute MUST be present if the item is an archived item. Otherwise, this attribute is optional. If this attribute is not present, the value is assumed to equal "false". This attribute SHOULD <4> be included.

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
FolderId	xs:string [XMLSCHEMA2]	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
ItemId	xs:string [XMLSCHEMA2]	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:choice>
</xs:complexType>
```

```

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

3.1.4.1.4 ConvertId Simple Types

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

Simple type name	Description
IdFormatType	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: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
EwsLegacyId	Specifies that the identifier is in the format that is used by the legacy version of the protocol. MAY <5> be present.
EwsId	Specifies that the identifier is in the format that is used by the current version of the protocol.
EntryId	Specifies that the identifier is in the format that can be used with the PidTagEntryId property, as specified in [MS-OXPROPS] section 2.761.
HexEntryId	Specifies that the identifier is a hexadecimal-encoded representation of the format that can be used with the PidTagEntryId property.
StoreId	Specifies that the identifier is in a format that is recognized only by the server.
OwaId	Specifies that the identifier is in a format that is used by the Web-based client capability of the server, if one exists.

3.1.5 Server Timer Events

None.

3.1.6 Server Other Local Events

None.

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="RgAAAAAzFgecVl%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:m="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
JkNmMtNGIwYy1hZWfhLTcwNjYzMDNjYjlkZQBGAIAAAAzFgecVl/WTo7NTrcPscM3
BwDHxFWht3DVTTrtUQkRbTXDCAAAAMUgfaADHxFWht3DVTTrtUQkRbTXDCAAAAMW98AAA="
Mailbox="User1@example.com" /> </m:ConvertIdResponseMessage> </m:ResponseMessages>
</m:ConvertIdResponse> </s:Body></s:Envelope>
```

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.	7.1
MS-OXWSCVTID-types.xsd	Contains the XML schema type definitions that are used in this protocol.	7.2

These files have to be placed in a common folder for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-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="Exchange2010_SP2"
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=" Exchange2010_SP2"
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">
  <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="Impersonation" use="literal"/>
        <soap:header message="tns:ConvertIdSoapIn" part="RequestVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
          <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>
```


7 Appendix B: Full XML Schema

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

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

These files have to be placed in a common folder in order for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-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	[MS-OXWSCDATA] section 7.1
MS-OXWSCVTID-types.xsd	7.2

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
  elementFormDefault="qualified" version="Exchange2010_SP2" 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:schema>
```

```

</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:attribute name="DestinationFormat" type="t:IdFormatType"
use="required"/>
    </xs:extension>
  </xs:complexContent>
</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	[MS-OXWSCDATA] 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="Exchange2010_SP2" 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: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>

```

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 2010
- Microsoft® Exchange Server 15 Technical Preview
- Microsoft® Outlook® 2010
- Microsoft® Outlook® 15 Technical 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:](#) Outlook 2010 uses the **ConvertId** operation to convert an "EntryId" value to an "EwsId" value.

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

9 Change Tracking

This section identifies changes that were made to the [MS-OXWSCVTID] protocol document between the October 2011 and January 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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

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

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

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

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

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

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

Some important terms used in 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 protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
3.1.4.1.3.5 t:AlternateIdType Complex Type	Updated the description of the Id attribute to specify that the maximum length is after base64 decoding.	N	Content updated.
3.1.4.1.3.6 t:AlternatePublicFolderIdType Complex Type	Updated the description of the FolderId attribute to specify that the maximum length is after base64 decoding.	N	Content updated.
3.1.4.1.3.7 t:AlternatePublicFolderItemIdType Complex Type	Updated the description of the ItemId attribute to specify that the maximum length is after base64 decoding.	N	Content updated.
8 Appendix C: Product Behavior	Added Exchange 15 Technical Preview and Outlook 15 Technical Preview to the list of applicable product versions.	Y	Content updated.

10 Index

A

Abstract data model
[server](#) 11
[Applicability](#) 7
[Attribute groups](#) 10
[Attributes](#) 10

C

[Capability negotiation](#) 7
[Change tracking](#) 29
[Complex types](#) 10

D

Data model - abstract
[server](#) 11

E

Events
[local - server](#) 20
[timer - server](#) 20

F

[Fields - vendor-extensible](#) 7
[Full WSDL](#) 23

G

[Glossary](#) 5
[Groups](#) 10

I

[Implementer - security considerations](#) 22
[Index of security parameters](#) 22
[Informative references](#) 6
Initialization
[server](#) 11
[Introduction](#) 5

L

Local events
[server](#) 20

M

Messages
[attribute groups](#) 10
[attributes](#) 10
[complex types](#) 10
[elements](#) 10
[enumerated](#) 9
[groups](#) 10

[namespaces](#) 9
[simple types](#) 10
[syntax](#) 9
[transport](#) 9

N

[Namespaces](#) 9
[Normative references](#) 5

O

[Overview \(synopsis\)](#) 6

P

[Parameters - security index](#) 22
[Preconditions](#) 7
[Prerequisites](#) 7
[Product behavior](#) 28

R

[References](#) 5
[informative](#) 6
[normative](#) 5
[Relationship to other protocols](#) 6

S

Security
[implementer considerations](#) 22
[parameter index](#) 22
Server
[abstract data model](#) 11
[initialization](#) 11
[local events](#) 20
[timer events](#) 20
[timers](#) 11
[Simple types](#) 10
[Standards assignments](#) 8
Syntax
[messages - overview](#) 9

T

Timer events
[server](#) 20
Timers
[server](#) 11
[Tracking changes](#) 29
[Transport](#) 9
Types
[complex](#) 10
[simple](#) 10

V

[Vendor-extensible fields](#) 7
[Versioning](#) 7

W

[WSDL](#) 23

Preliminary