[MS-OXWSXPROP]:
Extended Properties Web Service Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
07/15/2009	1.0	Major	Initial Availability.
11/04/2009	1.1.0	Minor	Updated the technical content.
02/10/2010	2.0.0	Major	Updated and revised the technical content.
05/05/2010	2.0.1	Editorial	Revised and edited the technical content.
08/04/2010	3.0	Major	Significantly changed the technical content.
11/03/2010	4.0	Major	Significantly changed the technical content.
03/18/2011	5.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Extended Properties Web Service Protocol is used by clients to manipulate extended properties on various types of objects. Extended properties are custom properties that users set on items and folders in a server's mailbox.

Sections 1.7 and 2 of this specification are normative and 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]:

GUID
XML

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

Web Services Description Language (WSDL)
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

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

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

[MS-OXWSFOLD] Microsoft Corporation, "Folders and Folder Permissions Web Service Protocol Specification", July 2009.

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

[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] World Wide Web Consortium, "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation 8 December 2009, http://www.w3.org/TR/REC-xml-names/

[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", March 2007.

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", April 2008.

1.3 Overview

This protocol specifies the extended property structure used by protocols that handle items and folders.

1.4 Relationship to Other Protocols

This protocol uses operations that are part of the Core Items Web Service Protocol, as described in [MS-OXWSCORE], and the Folder and Folder Permissions Web Service Protocol, as described in [MS-OXWSFOLD].

Extended properties can be added or updated as specified in the following specifications for these operations:

- The **CreateItem** operation, as specified in [MS-OXWSCORE] section 3.1.4.2.
- The **UpdateItem** operation, as specified in [MS-OXWSCORE] section 3.1.4.7.
- The **CreateFolder** operation, as specified in [MS-OXWSFOLD] section 3.1.4.2.
- The **UpdateFolder** operation, as specified in[MS-OXWSFOLD] section 3.1.4.8.

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

The protocol information provided in this document applies to protocols that use extended properties.

1.7 Versioning and Capability Negotiation

This document specifies versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with SOAP 1.1, as specified in section 2.1.
- **Protocol Versions:** This protocol has only one **WSDL port type** version. The **WSDL** version of the request is identified by using the **t:RequestServerVersion** element, as described in [MS-

OXWSCDATA] section 2.2.4.7, and the version of the server responding to the request is identified by using the **t:ServerVersionInfo** element, as described in [MS-OXWSCDATA] section 2.2.4.8.

- **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

2.1 Transport

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

2.2 Common Message Syntax

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

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
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]
targetNamespace	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 XML schema message definitions.

2.2.3 Elements

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

Element name	Description
ExtendedFieldURI	Specifies an extended MAPI property.

2.2.3.1 ExtendedFieldURI Element

The **ExtendedFieldURI** element specifies an extended MAPI property.

```
<xs:element name="ExtendedFieldURI"
  type="t:PathToExtendedFieldType"
  substitutionGroup="t:Path"
```

```
/>
```

This element allows substitution of the **Path** element, as specified in [MS-OXWSCDATA] section 2.2.4.6), for the **PathToExtendedFieldType** element (section 2.2.4.3).

2.2.4 Complex Types

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

Complex type name	Description
NonEmptyArrayOfPropertyValuesType	Represents a collection of values for an extended property.
ExtendedPropertyType	Represents extended properties on folders and items.
PathToExtendedFieldType	Represents the definition of an extended property.

2.2.4.1 t:NonEmptyArrayOfPropertyValuesType Complex Type

The **NonEmptyArrayOfPropertyValuesType** complex type represents a collection of values for an extended property. This array has at least one member.

```
<xs:complexType name="NonEmptyArrayOfPropertyValuesType">
  <xs:choice>
    <xs:element name="Value"
      type="xs:string"
      maxOccurs="unbounded"
     />
  </xs:choice>
</xs:complexType>
```

The following table lists the child element of the **NonEmptyArrayOfPropertyValuesType** complex type.

Element name	Type	Description
Value	**xs:string** [XMLSCHEMA2]	Specifies a value for an extended property.

2.2.4.2 t:ExtendedPropertyType Complex Type

The **ExtendedPropertyType** complex type specifies extended properties on folders and items. This complex type represents an extended property instance, including both its path identifier and its associated value.

```
<xs:complexType name="ExtendedPropertyType">
  <xs:sequence>
    <xs:element name="ExtendedFieldURI"
      type="t:PathToExtendedFieldType"
     />
```

```
    <xs:choice>
      <xs:element name="Value"
        type="xs:string"
       />
      <xs:element name="Values"
        type="t:NonEmptyArrayOfPropertyValuesType"
       />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

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

Element name	Type	Description
ExtendedFieldURI	**t:PathToExtendedFieldType (**section 2.2.4.3**)**	Specifies the path to the extended property.
Value	**xs:string** [XMLSCHEMA2]	Specifies a single value for the property specified by the **ExtendedFieldURI** element (section 2.2.3.1).
Values	**t:NonEmptyArrayOfPropertyValuesType (**section 2.2.4.1**)**	Specifies two or more values for the property specified by the **ExtendedFieldURI** element (section 2.2.3.1).

2.2.4.3 t:PathToExtendedFieldType Complex Type

The **PathToExtendedFieldType** complex type specifies an extended property. The **PathToExtendedFieldType** complex type extends the **BasePathToElementType** complex type ([MS-OXWSCDATA] section 2.2.3.13).

```
<xs:complexType name="PathToExtendedFieldType">
  <xs:complexContent>
    <xs:extension
      base="t:BasePathToElementType"
    >
      <xs:attribute name="DistinguishedPropertySetId"
        type="t:DistinguishedPropertySetType"
        use="optional"
       />
      <xs:attribute name="PropertySetId"
        type="t:GuidType"
        use="optional"
       />
      <xs:attribute name="PropertyTag"
        type="t:PropertyTagType"
        use="optional"
       />
      <xs:attribute name="PropertyName"
        type="xs:string"
        use="optional"
       />
      <xs:attribute name="PropertyId"
```

```
        type="xs:int"
        use="optional"
       />
      <xs:attribute name="PropertyType"
        type="t:MapiPropertyTypeType"
        use="required"
       />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table summarizes the set of common XML schema attribute definitions defined by this specification.

Attribute name	Type	Description
DistinguishedPropertySetId	**t:DistinguishedPropertySetType** ([MS-OXWSCDATA] section 2.2.2.10)	Specifies well-known property set IDs for extended properties. If this attribute is used, the **PropertySetId** and **PropertyTag** attributes cannot be used. This attribute MUST be used with the **PropertyType** attribute and either the **PropertyId** or **PropertyName** attribute.
PropertySetId	**t:GuidType** (section 2.2.5.1)	Specifies an extended property set or namespace by its identifying **GUID**. If this attribute is used, the **DistinguishedPropertySetId** and **PropertyTag** attributes cannot be used. This attribute MUST be used with the **PropertyType** attribute and either the **PropertyId** or **PropertyName** attribute,.
PropertyTag	**t:PropertyTagType** (section 2.2.5.3)	Specifies the property tag. The **PropertyTag** attribute can be represented as either a hexadecimal value or a short integer. If the **PropertyTag** attribute is used, the **DistinguishedPropertySetId**, **PropertySetId**, **PropertyName**, and **PropertyId** attributes MUST NOT be used.
PropertyName	**xs:string** [XMLSCHEMA2]	Specifies an extended property by its name. This property MUST be coupled with either the **DistinguishedPropertySetId** or **PropertySetId** attribute.
PropertyId	**xs:int** [XMLSCHEMA2]	Specifies an extended property

Attribute name	Type	Description
		by its dispatch ID. This property MUST be coupled with either the **DistinguishedPropertySetId** or **PropertySetId** attribute.
PropertyType	**t:MapiPropertyTypeType** (section 2.2.5.2)	Specifies the property type of an extended property.

For examples that show how the **ExtendedFieldURI** element, the **PropertySetId**, **PropertyName**, and **PropertyType** attributes, and the property values are created and retrieved along with the associated message, see sections 4.1 and 4.2.

2.2.5 Simple Types

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

Simple type name	Description
GuidType	Specifies an extended property set or namespace by its identifying GUID.
MapiPropertyTypeType	Specifies the property type.
PropertyTagType	Specifies the property tag.

2.2.5.1 t:GuidType Simple Type

The **GuidType** simple type specifies an extended property set or namespace by its identifying GUID.

```
<xs:simpleType name="GuidType">
  <xs:restriction
    base="xs:string"
  >
    <xs:pattern
      value="[0-9A-Fa-f]{8}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{12}"
     />
  </xs:restriction>
</xs:simpleType>
```

The following pattern is defined by the **GuidType** simple type:

```
[0-9A-Fa-f]{8}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{12}
```

2.2.5.2 t:MapiPropertyTypeType Simple Type

The **MapiPropertyTypeType** simple type specifies the property type.

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

```
    >
      <xs:enumeration
        value="ApplicationTime"
       />
      <xs:enumeration
        value="ApplicationTimeArray"
       />
      <xs:enumeration
        value="Binary"
       />
      <xs:enumeration
        value="BinaryArray"
       />
      <xs:enumeration
        value="Boolean"
       />
      <xs:enumeration
        value="CLSID"
       />
      <xs:enumeration
        value="CLSIDArray"
       />
      <xs:enumeration
        value="Currency"
       />
      <xs:enumeration
        value="CurrencyArray"
       />
      <xs:enumeration
        value="Double"
       />
      <xs:enumeration
        value="DoubleArray"
       />
      <xs:enumeration
        value="Error"
       />
      <xs:enumeration
        value="Float"
       />
      <xs:enumeration
        value="FloatArray"
       />
      <xs:enumeration
        value="Integer"
       />
      <xs:enumeration
        value="IntegerArray"
       />
      <xs:enumeration
        value="Long"
       />
      <xs:enumeration
        value="LongArray"
       />
      <xs:enumeration
        value="Null"
       />
      <xs:enumeration
```

```
        value="Object"
       />
      <xs:enumeration
        value="ObjectArray"
       />
      <xs:enumeration
        value="Short"
       />
      <xs:enumeration
        value="ShortArray"
       />
      <xs:enumeration
        value="SystemTime"
       />
      <xs:enumeration
        value="SystemTimeArray"
       />
      <xs:enumeration
        value="String"
       />
      <xs:enumeration
        value="StringArray"
       />
    </xs:restriction>
  </xs:simpleType>
```

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

Value	Description
ApplicationTime	Specifies a double value that is interpreted as a date and time. The integer part is the date, and the fraction part is the time.
ApplicationTimeArray	Specifies an array of double values that are interpreted as a date and time.
Binary	Specifies a base64-encoded binary value.
BinaryArray	Specifies an array of base64-encoded binary values.
Boolean	Specifies a Boolean "true" or "false" value.
CLSID	Specifies a GUID string.
CLSIDArray	Specifies an array of GUID strings.
Currency	Specifies a 64-bit integer that is interpreted as the number of cents.
CurrencyArray	Specifies an array of 64-bit integers that are interpreted as the number of cents.
Double	Specifies a 64-bit floating-point value.
DoubleArray	Specifies an array of 64-bit floating-point values.
Error	Specifies an SCODE value; this is a 32-bit unsigned integer. This value is not used for restrictions or for getting or setting values. This value exists only for reporting.

Value	Description
Float	Specifies a 32-bit floating-point value.
FloatArray	Specifies an array of 32-bit floating-point values.
Integer	Specifies a signed 32-bit (**Int32**) integer.
IntegerArray	Specifies an array of signed 32-bit (**Int32**) integers.
Long	Specifies a signed or unsigned 64-bit (**Int64**) integer.
LongArray	Specifies an array of signed or unsigned 64-bit (**Int64**) integers.
Null	Indicates no property value. This value is not used for restrictions or for getting or setting values. This value exists only for reporting.
Object	Specifies a pointer to an object that implements the **IUnknown** interface. This value is not used for restrictions or for getting or setting values. This value exists only for reporting.
ObjectArray	Specifies an array of pointers to objects that implement the **IUnknown** interface. This value is not used for restrictions or for getting or setting values. This value exists only for reporting.
Short	Specifies a signed 16-bit integer.
ShortArray	Specifies an array of signed 16-bit integers.
SystemTime	Specifies a 64-bit integer date and time value in the form of a **FILETIME** structure.
SystemTimeArray	Specifies an array of 64-bit integer date and time values in the form of a **FILETIME** structure.
String	Specifies a Unicode string.
StringArray	Specifies an array of Unicode strings.

2.2.5.3 t:PropertyTagType Simple Type

The **PropertyTagType** simple type specifies the property tag.

```
<xs:simpleType name="PropertyTagType">
  <xs:union
    memberTypes="xs:unsignedShort"
  >
    <xs:simpleType
      id="HexPropertyTagType"
    >
      <xs:restriction
        base="xs:string"
      >
        <xs:pattern
          value="(0x|0X)[0-9A-Fa-f]{1,4}"
         />
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
```

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

The following pattern is defined by the **PropertyTagType** simple type:

```
(0x|0X)[0-9A-Fa-f]{1,4}
```

Note that the property tag can be represented in either hexadecimal or decimal form.

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3 Protocol Details

This document specifies common XML schema types and elements and does not provide any protocol details. This document also does not specify any operations.

3.1 Server Details

This document specifies common XML schema types and elements and does not provide any server protocol details.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

This protocol includes the operations that are listed in the following table.

Operation	Description
CreateItem	The **CreateItem** operation creates items on the server. This operation is specified in [MS-OXWSCORE] section 3.1.4.2.
UpdateItem	The **UpdateItem** operation updates items on the server. This operation is specified in [MS-OXWSCORE] section 3.1.4.7.
CreateFolder	The **CreateFolder** operation creates folders on the server. This operation is specified in [MS-OXWSFOLD] section 3.1.4.2.
UpdateFolder	The **UpdateFolder** operation modifies properties of a folder on the server. This operation is specified in [MS-OXWSFOLD] section 3.1.4.8.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following examples show the request and response **XML** that is used to perform the specified operations.

4.1 Create Extended Properties Example

The following example creates and sends mail with three extended properties.

The client constructs the request XML and sends the newly created message to the server. The message has three custom properties. The first custom property is named **Expiration Date** and has a string value set to a time of "12/25/2009 3:25:15 PM". The second custom property is named **Employee Type** and has a string value set to "Part Time". The third custom property is named **MyFlag** and has an integer value set to "4". Each of these extended properties becomes part of the message. The **PropertySetId** attribute values are **GUIDs**, and are shown in these examples.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010" />
  </soap:Header>
  <soap:Body>
    <m:CreateItem MessageDisposition="SendAndSaveCopy">
      <m:SavedItemFolderId>
        <t:DistinguishedFolderId Id="sentitems" />
      </m:SavedItemFolderId>
      <m:Items>
        <t:Message>
          <t:Subject>Interesting</t:Subject>
          <t:Body BodyType="HTML">The merger is finalized.</t:Body>
          <t:ExtendedProperty>
            <t:ExtendedFieldURI PropertySetId="c11ff724-aa03-4555-9952-8fa248a11c3e"
               PropertyName="Expiration Date" PropertyType="String" />
            <t:Value>12/25/2009 3:25:15 PM</t:Value>
          </t:ExtendedProperty>
          <t:ExtendedProperty>
            <t:ExtendedFieldURI PropertySetId="24a3075f-a8b7-4181-a9ed-708a947b8765"
               PropertyName="Employee Type" PropertyType="String" />
            <t:Value>Part Time</t:Value>
          </t:ExtendedProperty>
          <t:ExtendedProperty>
            <t:ExtendedFieldURI PropertySetId="75a5486f-9267-49ca-9b4e-3d04ca9ec179"
               PropertyName="MyFlag" PropertyType="Integer" />
            <t:Value>4</t:Value>
          </t:ExtendedProperty>
          <t:ToRecipients>
            <t:Mailbox>
              <t:EmailAddress>User1@Contoso.com</t:EmailAddress>
            </t:Mailbox>
            <t:Mailbox>
              <t:EmailAddress>User2@Contoso.com</t:EmailAddress>
            </t:Mailbox>
          </t:ToRecipients>
        </t:Message>
```

```
        </m:Items>
      </m:CreateItem>
    </soap:Body>
  </soap:Envelope>
```

The server constructs the response XML and sends it to the client, as shown in the following example.

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

4.2 Retrieving Extended Properties Example

The following example gets a message and retrieves the three specified extended properties.

The client constructs the request XML and sends it to the server. The client is requesting the specified message with the three named extended properties — **Expiration Date**, **Employee Type**, and **MyFlag**. The **PropertySetId** attribute values are GUIDs, and are shown in these examples.. Note that the **ItemId** and **ChangeKey** attributes have been shortened to preserve readability.

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

```
          <t:BaseShape>IdOnly</t:BaseShape>
          <t:AdditionalProperties>
            <t:ExtendedFieldURI PropertySetId="c11ff724-aa03-4555-9952-8fa248a11c3e"
PropertyName="Expiration Date" PropertyType="String" />
            <t:ExtendedFieldURI PropertySetId="24a3075f-a8b7-4181-a9ed-708a947b8765"
PropertyName="Employee Type" PropertyType="String" />
            <t:ExtendedFieldURI PropertySetId="75a5486f-9267-49ca-9b4e-3d04ca9ec179"
PropertyName="MyFlag" PropertyType="Integer" />
          </t:AdditionalProperties>
        </m:ItemShape>
        <m:ItemIds>
          <t:ItemId Id="AAMkAGIwODEy" ChangeKey="CQAAABYAAA" />
        </m:ItemIds>
      </m:GetItem>
    </soap:Body>
  </soap:Envelope>
```

The server constructs the response XML and sends it to the client. In this example, three extended properties are returned; these are the custom properties that were generated when the message was created. The **PropertySetId**, **PropertyName**, and **PropertyType** attributes are included in the response.

```
  <?xml version="1.0" encoding="utf-8"?>
  <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Header>
      <h:ServerVersionInfo MajorVersion="14" MinorVersion="1" MajorBuildNumber="63"
MinorBuildNumber="0"
          Version="Exchange2010_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:GetItemResponse
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
          xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
        <m:ResponseMessages>
          <m:GetItemResponseMessage ResponseClass="Success">
            <m:ResponseCode>NoError</m:ResponseCode>
            <m:Items>
              <t:Message>
                <t:ItemId Id="AAMkAGIwODEy" ChangeKey="CQAAABYAAA" />
                <t:ExtendedProperty>
                  <t:ExtendedFieldURI PropertySetId="c11ff724-aa03-4555-9952-8fa248a11c3e"
                     PropertyName="Expiration Date" PropertyType="String" />
                  <t:Value>12/25/2009 3:25:15 PM </t:Value>
                </t:ExtendedProperty>
                <t:ExtendedProperty>
                  <t:ExtendedFieldURI PropertySetId="24a3075f-a8b7-4181-a9ed-708a947b8765"
                     PropertyName="Employee Type" PropertyType="String" />
                  <t:Value>Part Time</t:Value>
                </t:ExtendedProperty>
                <t:ExtendedProperty>
                  <t:ExtendedFieldURI PropertySetId="75a5486f-9267-49ca-9b4e-3d04ca9ec179"
                     PropertyName="MyFlag" PropertyType="Integer" />
                  <t:Value>4</t:Value>
```

```
                </t:ExtendedProperty>
              </t:Message>
            </m:Items>
          </m:GetItemResponseMessage>
        </m:ResponseMessages>
      </m:GetItemResponse>
    </s:Body>
  </s:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

The following XML file is required to implement the functionality described in other protocol specifications. The contents of the file are contained in this section. This protocol does not define a WSDL file. This protocol defines an XML schema file that is referenced by other WSDL and XML schema files.

Protocol file name	Description	Section
MS-OXWSXPROP-types.xsd	Contains the XML schema type definitions used in this protocol.	section 6.1

This file have to be placed in a common folder for the WSDL or XML schema file to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSXPROP-types.xsd schema have to be placed in the same common folder as the file named in the table.

6.1 Types Schema

This section contains the MS-OXWSXPROP-types.xsd file and information about additional files that this schema file requires for correct operation.

MS-OXWSXPROP-types.xsd includes the file listed in the following table. To operate correctly, this file has to be present in the folder that contains the WSDL, types, and messages schema files.

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

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema 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/types"
elementFormDefault="qualified" version="Exchange2010" id="types">
     <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
     <xs:element name="ExtendedFieldURI" type="t:PathToExtendedFieldType"
substitutionGroup="t:Path"/>
     <xs:complexType name="ExtendedPropertyType">
          <xs:sequence>
               <xs:element name="ExtendedFieldURI" type="t:PathToExtendedFieldType"/>
               <xs:choice>
                    <xs:element name="Value" type="xs:string"/>
                    <xs:element name="Values" type="t:NonEmptyArrayOfPropertyValuesType"/>
               </xs:choice>
          </xs:sequence>
     </xs:complexType>
     <xs:simpleType name="GuidType">
       <xs:restriction base="xs:string">
         <xs:pattern value="[0-9A-Fa-f]{8}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{4}-[0-9A-Fa-f]{4}-[0-
9A-Fa-f]{12}"/>
       </xs:restriction>
     </xs:simpleType>
     <xs:simpleType name="MapiPropertyTypeType">
          <xs:restriction base="xs:string">
               <xs:enumeration value="ApplicationTime"/>
               <xs:enumeration value="ApplicationTimeArray"/>
               <xs:enumeration value="Binary"/>
               <xs:enumeration value="BinaryArray"/>
```

```
                <xs:enumeration value="Boolean"/>
                <xs:enumeration value="CLSID"/>
                <xs:enumeration value="CLSIDArray"/>
                <xs:enumeration value="Currency"/>
                <xs:enumeration value="CurrencyArray"/>
                <xs:enumeration value="Double"/>
                <xs:enumeration value="DoubleArray"/>
                <xs:enumeration value="Error"/>
                <xs:enumeration value="Float"/>
                <xs:enumeration value="FloatArray"/>
                <xs:enumeration value="Integer"/>
                <xs:enumeration value="IntegerArray"/>
                <xs:enumeration value="Long"/>
                <xs:enumeration value="LongArray"/>
                <xs:enumeration value="Null"/>
                <xs:enumeration value="Object"/>
                <xs:enumeration value="ObjectArray"/>
                <xs:enumeration value="Short"/>
                <xs:enumeration value="ShortArray"/>
                <xs:enumeration value="SystemTime"/>
                <xs:enumeration value="SystemTimeArray"/>
                <xs:enumeration value="String"/>
                <xs:enumeration value="StringArray"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:complexType name="NonEmptyArrayOfPropertyValuesType">
            <xs:choice>
                <xs:element name="Value" type="xs:string" maxOccurs="unbounded"/>
            </xs:choice>
        </xs:complexType>
        <xs:complexType name="PathToExtendedFieldType">
            <xs:complexContent>
                <xs:extension base="t:BasePathToElementType">
                    <xs:attribute name="DistinguishedPropertySetId"
    type="t:DistinguishedPropertySetType" use="optional"/>
                    <xs:attribute name="PropertySetId" type="t:GuidType" use="optional"/>
                    <xs:attribute name="PropertyTag" type="t:PropertyTagType"
    use="optional"/>
                    <xs:attribute name="PropertyName" type="xs:string" use="optional"/>
                    <xs:attribute name="PropertyId" type="xs:int" use="optional"/>
                    <xs:attribute name="PropertyType" type="t:MapiPropertyTypeType"
    use="required"/>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:simpleType name="PropertyTagType">
            <xs:union memberTypes="xs:unsignedShort">
                <xs:simpleType id="HexPropertyTagType">
                    <xs:restriction base="xs:string">
                        <xs:pattern value="(0x|0X)[0-9A-Fa-f]{1,4}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:union>
        </xs:simpleType>

    </xs:schema>
```

7 Appendix B: Product Behavior

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

- Microsoft® Exchange Server 2007
- Microsoft® Exchange Server 2010

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.

8 Change Tracking

This section identifies changes that were made to the [MS-OXWSXPROP] protocol document between the November 2010 and March 2011 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
1 Introduction	Added information about which sections of the specification are normative and can contain RFC 2119 language.	Y	Content updated for template compliance.

9 Index

T

V

W