[MS-OXWSXPROP]:

Extended Properties Structure

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
7/15/2009	1.0	Major	Initial Availability.
11/4/2009	1.1.0	Minor	Updated the technical content.
2/10/2010	2.0.0	Major	Updated and revised the technical content.
5/5/2010	2.0.1	Editorial	Revised and edited the technical content.
8/4/2010	3.0	Major	Significantly changed the technical content.
11/3/2010	4.0	Major	Significantly changed the technical content.
3/18/2011	5.0	Major	Significantly changed the technical content.
8/5/2011	6.0	Major	Significantly changed the technical content.
10/7/2011	6.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	7.0	Major	Significantly changed the technical content.
4/27/2012	7.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	7.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	8.0	Major	Significantly changed the technical content.
2/11/2013	8.0	None	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	8.0	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	8.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	8.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	9.0	Major	Significantly changed the technical content.
7/31/2014	9.0	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	9.1	Minor	Clarified the meaning of the technical content.
5/26/2015	10.0	Major	Significantly changed the technical content.
9/14/2015	10.0	None	No changes to the meaning, language, or formatting of the technical content.
6/13/2016	10.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	10.0	None	No changes to the meaning, language, or formatting of the technical content.
7/24/2018	11.0	Major	Significantly changed the technical content.

Date	Revision History	Revision Class	Comments
10/1/2018	12.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Extended Properties Structure 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. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

Messaging Application Programming Interface (MAPI): A messaging architecture that enables multiple applications to interact with multiple messaging systems across a variety of hardware platforms.

XML: The Extensible Markup Language, as described in [XML1.0].

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

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

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

1.2 References

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

1.2.1 Normative References

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

[MS-DCOM] Microsoft Corporation, "Distributed Component Object Model (DCOM) Remote Protocol".

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-OAUT] Microsoft Corporation, "OLE Automation Protocol".

[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, https://www.rfc-editor.org/rfc/rfc2119.html

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, https://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, https://www.w3.org/TR/2009/REC-xml-names-20091208/

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, https://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, https://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 Informative References

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

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

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

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

1.3 Overview

The Extended Properties Structure is used by protocols that handle items and folders.

1.4 Relationship to Protocols and Other Structures

This structure is used by 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 by using the following operations:

- The **CreateItem** operation, as described in [MS-OXWSCORE] section 3.1.4.2. The **CreateItem** operation creates items on the server.
- The **UpdateItem** operation, as described in [MS-OXWSCORE] section 3.1.4.9. The **UpdateItem** operation updates items on the server.
- The **GetItem** operation, as specified in [MS-OXWSCORE] section 3.1.4.4. The **GetItem** operation gets items on the server.
- The **CreateFolder** operation, as described in [MS-OXWSFOLD] section 3.1.4.2. The **CreateFolder** operation creates folders on the server.
- The **UpdateFolder** operation, as described in [MS-OXWSFOLD] section 3.1.4.8. The **UpdateFolder** operation modifies properties of a folder on the server.
- The **FindItem** operation, as described in [MS-OXWSSRCH] section 3.1.4.2. The **Findtem** operation finds items on the server.

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

1.5 Applicability Statement

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

1.6 Versioning and Localization

This specification covers versioning issues in the following areas:

- **Structure Versions**: This structure specifies the only version of the Extended Properties Structure.
- **Localization:** This structure specifies no locale-specific processes or data.

1.7 Vendor-Extensible Fields

None.

2 Structures

2.1 Extended Properties

The following sections specify the namespaces, elements, complex types, and simple types of the Extended Properties Structure. The elements and types use type definitions that are specified in [XMLSCHEMA1] and [XMLSCHEMA2].

2.1.1 Extended Properties 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
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
t	http://schemas.microsoft.com/exchange/services/2006/types	
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

2.1.2 ExtendedFieldURI Element

The **ExtendedFieldURI** element specifies an extended **Messaging Application Programming Interface (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 2.2.3.8, for the **PathToExtendedFieldType** complex type as specified in section 2.1.6.

2.1.3 NonEmptyArrayOfExtendedPropertyType Complex Type

The **NonEmptyArrayOfExtendedPropertyType** complex type represents an extended property. This array has at least one member.<1>

```
<xs:complexType name="NonEmptyArrayOfExtendedPropertyType">
  <xs:choice>
    <xs:element name="ExtendedProperty" type="t:ExtendedPropertyType" maxOccurs="unbounded"/>
  </xs:choice>
</xs:complexType>
```

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

Element name	Type	Description
ExtendedProperty	**ExtendedPropertyType** (section 2.1.5)	Specifies an extended property instance, including both its path identifier and its associated value.

2.1.4 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.

2.1.5 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**	Specifies the path to the extended property.
Value	**xs:string** ([XMLSCHEMA2])	Specifies a single value for the property specified by the **ExtendedFieldURI** element.
Values	**t:NonEmptyArrayOfPropertyValuesType** (section 2.1.4)	Specifies two or more values for the property specified by the **ExtendedFieldURI** element.

2.1.6 t:PathToExtendedFieldType Complex Type

The **PathToExtendedFieldType** complex type specifies an extended property. The **PathToExtendedFieldType** complex type extends the **BasePathToElementType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.16.

```
<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	Specifies well-known property set IDs for extended properties. If this attribute is used, the **PropertySetId** and

Attribute name	Type	Description
	2.2.5.11)	**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 2.1.7)	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 2.1.9)	Specifies the property tag. The **PropertyTag** attribute can be represented as either a hexadecimal value or a decimal value. 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 attribute MUST be coupled with either the **DistinguishedPropertySetId** or **PropertySetId** attribute.
PropertyId	**xs:int** ([XMLSCHEMA2])	Specifies an extended property by its dispatch ID. This attribute MUST be coupled with either the **DistinguishedPropertySetId** or **PropertySetId** attribute.
PropertyType	**t:MapiPropertyTypeType** (section 2.1.8)	Specifies the property type of an extended property.

2.1.7 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}
```

2.1.8 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 ([MS-OAUT] section 2.2.48) 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.
Float	Specifies a 32-bit floating-point value.
FloatArray	Specifies an array of 32-bit floating-point values.

Value	Description
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 ([MS-DCOM] section 3.1.1.5.8). 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** ([MS-DTYP] section 2.3.3) 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.

2.1.9 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.

3 Structure Examples

The following examples show how extended properties are created and retrieved.

3.1 Creating Extended Properties

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 **GUID**s, and are shown in the following 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="Exchange2013" />
  </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="Exchange2013"
        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>
```

3.2 Retrieving Extended Properties

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 **GUID**s, and are shown in the following 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="Exchange2013" />
  </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="Exchange2013"
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>
```

4 Security

4.1 Security Considerations for Implementers

None.

4.2 Index of Security Fields

None.

5 Appendix A: Full XML Schema

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 types schema for this structure.

File name	Defining specification
MS-OXWSCDATA-types.xsd	[MS-OXWSCDATA] section 7.2

The following sections specify the namespaces, elements, complex types, and simple types of the Extended Properties Structure. The elements and types use type definitions that are specified in [XMLSCHEMA1] and [XMLSCHEMA2].

```
<?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="Exchange2016" 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="NonEmptyArrayOfExtendedPropertyType">
    <xs:choice>
      <xs:element name="ExtendedProperty" type="t:ExtendedPropertyType"
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>
```

6 Appendix B: Product Behavior

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

- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Exchange Server 2019

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

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

<1> Section 2.1.3: Exchange 2007 and Exchange 2010 do not use the **NonEmptyArrayOfExtendedPropertyType** complex type.

7 Change Tracking

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

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

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

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

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

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

Section	Description	Revision class
6 Appendix B: Product Behavior	Updated list of supported products.	Major

8 Index