

[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.

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	6
1.6 Applicability Statement	6
1.7 Versioning and Capability Negotiation	6
1.8 Vendor Extensible Fields	6
1.9 Standards Assignments	7
2 Messages	8
2.1 Transport	8
2.2 Common Message Syntax	8
2.2.1 Namespaces	8
2.2.2 Messages	8
2.2.3 Elements	8
2.2.3.1 ExtendedFieldURI Element	8
2.2.4 Complex Types	9
2.2.4.1 t:NonEmptyArrayOfPropertyType Complex Type	9
2.2.4.2 t:ExtendedPropertyType Complex Type	9
2.2.4.3 t:PathToExtendedFieldType Complex Type	10
2.2.5 Simple Types	12
2.2.5.1 t:GuidType Simple Type	12
2.2.5.2 t:MapiPropertyTypeType Simple Type	12
2.2.5.3 t:PropertyTagType Simple Type	15
2.2.6 Attributes	16
2.2.7 Groups	16
2.2.8 Attribute Groups	16
3 Protocol Details	17
3.1 Server Details	17
3.1.1 Abstract Data Model	17
3.1.2 Timers	17
3.1.3 Initialization	17
3.1.4 Message Processing Events and Sequencing Rules	17
3.1.5 Timer Events	17
3.1.6 Other Local Events	17
3.2 Client Details	17
3.2.1 Abstract Data Model	17
3.2.2 Timers	17
3.2.3 Initialization	17
3.2.4 Message Processing Events and Sequencing Rules	17
3.2.5 Timer Events	18
3.2.6 Other Local Events	18
4 Protocol Examples	19
4.1 Create Extended Properties Example	19

4.2 Retrieving Extended Properties Example	20
5 Security	23
5.1 Security Considerations for Implementers	23
5.2 Index of Security Parameters	23
6 Appendix A: Full WSDL	24
6.1 Types Schema	24
7 Appendix B: Product Behavior	26
8 Change Tracking.....	27
9 Index	30

1 Introduction

This document specifies the Extended Properties Web Service protocol, which is used by clients to manipulate extended properties on an object. Extended properties are custom properties on items and folders in a server's mailbox.

1.1 Glossary

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

GUID
Web Services Description Language (WSDL)
WSDL message
WSDL port type
XML
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", W3C Note, 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., Eds., et al., "Namespaces in XML 1.0 (Third Edition)", December 2009, <http://www.w3.org/TR/REC-xml-names/>

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

1.2.2 Informative References

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

1.3 Overview

The Extended Web Service Properties protocol specifies the extended property structure used by protocols that handle items and folders.

1.4 Relationship to Other Protocols

Clients that implement this protocol use operations from the protocols listed in the following table to perform work.

Protocol	Description
Core Items Web Service protocol [MS-OXWSCORE]	Clients can use the CreateItem operation (section 3.1.4.2) and the UpdateItem operation (section 3.1.4.7) to create or update the item.
Folders and Folder Permissions Web Service protocol [MS-OXWSFOLD]	Clients can use the CreateFolder operation (section 3.1.4.2) and UpdateFolder operation (section 3.1.4.8) to add or update Extended Properties to a folder.

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

The protocol specifications in this document apply 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 SOAP 1.1, as specified in section [2.1](#).
- **Protocol Versions:** This protocol specifies only one **WSDL** portType version.
- **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	[MS-OXWSXPROP]
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
targetNamespace	http://schemas.microsoft.com/exchange/services/2006/messages	[MS-OXWSXPROP]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	[MS-OXWSXPROP]

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	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"
/>
```

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

Allows substitution of the <Path> element (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	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:NonEmptyArrayOfPropertyType 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>
```

Child Elements

Element	Type	Description
Value	xs:string	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.

```
<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>

```

Child Elements

Element	Type	Description
ExtendedFieldURI	t:PathToExtendedFieldType	Specifies the path to the extended property.
Value	xs:string	Specifies a single value for the property specified by the <ExtendedFieldURI> element.
Values	t:NonEmptyArrayOfPropertyValuesType	Specifies two or more values for the property specified by the <ExtendedFieldURI> element.

Represents an extended property instance, both its path identifier along with its associated value.

2.2.4.3 t:PathToExtendedFieldType Complex Type

The **PathToExtendedFieldType** complex type specifies an extended property.

```

<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>

```

Attributes

Name	Type	Description
DistinguishedPropertySetId	t:DistinguishedPropertySetType	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 either the PropertyId or PropertyName attributes, and the PropertyType attribute.
PropertySetId	t:GuidType	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 either the PropertyId or PropertyName attribute, and the PropertyType attribute.
PropertyTag	t:PropertyTagType	Specifies the property tag. The PropertyTag attribute can be represented as either a hexadecimal or a short integer. If the PropertyTag attribute is used, the DistinguishedPropertySetId , PropertySetId , PropertyName , and PropertyId attributes MUST NOT be used.
PropertyName	xs:string	Specifies an extended property by its name. This property MUST be coupled with either the DistinguishedPropertySetId or the PropertySetId attribute.
PropertyId	xs:int	Specifies an extended property by its dispatch ID. This property MUST be coupled with either the DistinguishedPropertySetId or the PropertySetId attribute. For a list of possible types, see section 2.2.5.2 .

Name	Type	Description
PropertyType	t:MapiPropertyTypeType	Specifies the property type of an extended property.

See sections [4.1](#) and [4.2](#) for examples that show how the <ExtendedFieldURI> element, the **PropertySetId**, **PropertyName**, and **PropertyType** attributes, and the property value are created and retrieved along with the associated message.

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	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>
```

Patterns

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>

```

Enumeration

The following values are defined by the **MapiPropertyTypeType** simple type:

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

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

Patterns

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

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

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 have any protocol details. This document does not specify any operations.

3.1 Server Details

This document specifies common XML schema types and elements, and does not have 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

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Client Details

This document specifies common XML schema types and elements, and does not have any client protocol details.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Message Processing Events and Sequencing Rules

None.

3.2.5 Timer Events

None.

3.2.6 Other Local Events

None.

4 Protocol Examples

The following examples, [Create Extended Properties Example](#) and [Retrieving Extended Properties Example](#) show the request and response XML to perform the specific 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 it to the server. The newly created message is sent to the server. The message has three custom properties. The first custom property is named Expiration Date with a string value set to a time of 12/25/2009 3:25:15 PM. The second custom property is named Employee Type with a string value set to Part Time. The third custom property is named MyFlag with an integer value set to 4. Each of these extended properties becomes part of the message. The PropertySetIds are GUIDs and these are examples only.

```
<?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>
```

```

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

```

The server constructs the response XML and sends it to the client.

```

<?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 identified 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 PropertySetIds are GUIDs and these are examples only. 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>
      </m:ItemShape>
    </m:GetItem>
  </soap:Body>
</soap:Envelope>

```

```

    <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 extended properties are the custom properties that were generated when the message was created. The **PropertySetId**, **PropertyName**, and **PropertyType** attributes are included.

```

<?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>

```

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

5 Security

5.1 Security Considerations for Implementers

The Extended Properties Web Service protocol does not use any additional security mechanisms.

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.

Section	Protocol Filename	Description
Types Schema	MS-OXWSXPROP-types.xsd	Contains the XML schema type definitions used in this protocol.

This file **MUST** be placed in a common folder for any referencing WSDL or XML schema file to validate and operate. Also, any schema files that are included or imported into the MS-OXWSXPROP-types.xsd schema **MUST** be placed in the common folder with 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:restriction>
  </xs:simpleType>
</xs:schema>
```



```

        <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:

- Microsoft® Exchange Server 2010

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. 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 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 product does not follow the prescription.

8 Change Tracking

This section identifies changes that were made to the [MS-OXWSXPROP] protocol document between the May 2010 and August 2010 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
[MS-OXWSXPROP]: Extended Properties Web Service Protocol Specification	57266 Modified protocol name	N	Editorially updated.
1.1 Glossary	Added the term GUID to the list of terms that are defined in [MS-OXGLOS].	N	Content update.
1.2.1 Normative References	55751 Moved [MS-OXGLOS] from Normative References section to Informative References section.	N	Content update.
1.2.1 Normative References	57265 Added references to protocols that use Extended Properties.	N	New content added.
1.2.1 Normative References	Removed reference to [RFC2396], [RFC2616], and [RFC2818].	N	Content update.
1.4 Relationship to Other Protocols	57265 Added Extended Properties referencing operations.	N	New content added.
2.2.4 Complex Types	54727 Updated the descriptions for the complex types.	N	Content update.
2.2.4.1	54727	N	Content

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
t:NonEmptyArrayOfPropertyType Complex Type	Updated the description of the collection and updated the syntax to add the NonEmptyArrayOfPropertyValuesType complex type name.		update.
2.2.4.2 t:ExtendedPropertyType Complex Type	54727 Updated the description of the <ExtendedFieldURI> element to clarify that it includes the path to the extended property.	N	Content update.
2.2.4.3 t:PathToExtendedFieldType Complex Type	54727 Clarified the descriptions for the PropertyTag and Propertytype attributes.	N	Content update.
2.2.5 Simple Types	54727 Modified the description of the PropertyTagType simple type	N	Content removed.
2.2.5.2 t:MapIPropertyTypeType Simple Type	Updated the syntax.	N	Content update.
2.2.5.3 t:PropertyTagType Simple Type	54727 Updated the description of the PropertyTagType simple type.	N	Content removed.
5.1 Security Considerations for Implementers	57151 Updated the protocol name in the text.	N	Content update.
6.1 Types Schema	56414 Updated the schema to include the GuidType simple type.	Y	New content added.
6.1 Types Schema	56436 Updated the schema to remove the <Value> element.	N	Content update.
8 Change Tracking	55413 Restructured the content	N	Content update.

9 Index

A

[Applicability](#) 6

C

[Capability negotiation](#) 6

[Change tracking](#) 27

Client

[abstract data model](#) 17

[initialization](#) 17

[local events](#) 18

[message processing](#) 17

[overview](#) 17

[sequencing rules](#) 17

[timer events](#) 18

[timers](#) 17

F

[Full WSDL](#) 24

G

[Glossary](#) 5

I

[Introduction](#) 5

M

Messages

[syntax](#) 8

[transport](#) 8

O

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

P

[Preconditions](#) 6

[Prerequisites](#) 6

[Product Behavior](#) 26

R

[Relationship to other protocols](#) 6

S

Security

[implementer considerations](#) 23

[overview](#) 23

[parameter index](#) 23

Server

[abstract data model](#) 17

[initialization](#) 17

[local events](#) 17

[message processing](#) 17

[overview](#) 17

[sequencing rules](#) 17

[timer events](#) 17

[timers](#) 17

[Standards assignments](#) 7

T

[Tracking changes](#) 27

V

[Vendor-extensible fields](#) 6

[Versioning](#) 6