[MS-OXOUM]:
Voice Mail and Fax Objects Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names.** The example companies, organizations, products, domain names, email 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
04/04/2008	0.1	Major	Initial Availability
04/25/2008	0.2		Revised and updated property names and other technical content.
06/27/2008	1.0		Initial Release.
08/06/2008	1.01		Updated references to reflect date of initial release.
09/03/2008	1.02		Revised and edited technical content.
12/03/2008	1.03		Minor editorial fixes.
03/04/2009	1.04		Revised and edited technical content.
04/10/2009	2.0		Updated technical content and applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	3.0.1	Editorial	Revised and edited the technical content.
02/10/2010	3.1.1	Minor	Updated the technical content.
05/05/2010	3.2.0	Minor	Updated the technical content.
08/04/2010	3.3	Minor	Clarified the meaning of the technical content.
11/03/2010	3.3	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	4.0	Major	Significantly changed the technical content.
08/05/2011	4.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/07/2011	4.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	5.0	Major	Significantly changed the technical content.
04/27/2012	6.0	Major	Significantly changed the technical content.
07/16/2012	6.1	Minor	Clarified the meaning of the technical content.
10/08/2012	7.0	Major	Significantly changed the technical content.
02/11/2013	7.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/26/2013	8.0	Major	Significantly changed the technical content.
11/18/2013	8.0	No change	No changes to the meaning, language, or formatting of

Date	Revision History	Revision Class	Comments
			the technical content.
02/10/2014	8.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1   Introduction

The Voice Mail and Fax Objects Protocol enables servers to create and send **Unified Messaging** objects.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1   Glossary

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

XML namespace

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

binary large object (BLOB)
codec
Contact object
dictionary
header
mailbox
message class
Message object
Multipurpose Internet Mail Extensions (MIME)
recipient
rights-managed email message
Simple Mail Transfer Protocol (SMTP)
special folder
stream
Unified Messaging
Uniform Resource Locator (URL)
voice message
XML schema

The following terms are specific to this document:

fax message: A Message object that contains a digital representation of content received from a fax machine.

missed call notification: A Message object that is intended to convey information about a call that was missed. The Message object contains information about the calling party and the time of the call, but does not contain audio content.

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

1.2   References

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

1.2.1 Normative References

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

[ASF] Microsoft Corporation, "Advanced Systems Format Specification", December 2004, http://download.microsoft.com/download/7/9/0/790fecaa-f64a-4a5e-a430-0bccdab3f1b4/ASF_Specification.doc

[G711] ITU-T, "Pulse code modulation (PCM) of voice frequencies", Recommendation G.711, November 1988, http://www.itu.int/rec/T-REC-G.711-198811-I/en

[GSM610] ETSI, "European digital cellular telecommunications system (Phase 1); Full rate speech; Transcoding (GSM 06.10)", February 1992, http://pda.etsi.org/pda/home.asp?wki_id=v9jLO9Nb7wSVbbYKNyW

[MS-OXCDATA] Microsoft Corporation, "Data Structures".

[MS-OXCMAIL] Microsoft Corporation, "RFC 2822 and MIME to Email Object Conversion Algorithm".

[MS-OXCMSG] Microsoft Corporation, "Message and Attachment Object Protocol".

[MS-OXOCFG] Microsoft Corporation, "Configuration Information Protocol".

[MS-OXOMSG] Microsoft Corporation, "Email Object Protocol".

[MS-OXORMMS] Microsoft Corporation, "Rights-Managed Email Object Protocol".

[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol".

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

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

[WAVE] IBM Corporation and Microsoft Corporation, "Multimedia Programming Interface and Data Specifications 1.0", August 1991, http://www-mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/Docs/riffmci.pdf

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

[XMLSCHEMA2/2] Biron, P.V., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

1.2.2 Informative References

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

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

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

1.3 Overview

Unified Messaging objects are items created on behalf of telephone callers or fax senders by the server. These objects are stored in the called party's **mailbox** on the server.

The server creates three types of Unified Messaging objects: **voice messages**, **fax messages**, and **missed call notifications**.

1.4 Relationship to Other Protocols

The Voice Mail and Fax Objects Protocol relies on the Special Folders Protocol, which is described in [MS-OXOSFLD], and the Message and Attachment Object Protocol, which is described in [MS-OXCMSG].

The Voice Mail and Fax Objects Protocol uses the Message and Attachment Object Protocol as a transport protocol between the client and the server.

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

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

This protocol can be used to show the electronic equivalent of telephony-based messages, such as voice messages, fax messages, and missed call notifications.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

This protocol does not provide any extensibility beyond that specified in [MS-OXCMSG].

1.9 Standards Assignments

None.

2   Messages

2.1   Transport

The Voice Mail and Fax Objects Protocol uses the Message and Attachment Object Protocol, as specified in [MS-OXCMSG], to create and store the three types of Unified Messaging objects.

2.2   Message Syntax

Unlike many other client-server objects, the server creates Unified Messaging objects. The server MUST include the general properties, as specified in [MS-OXCMSG] section 2.2.1.1. The server SHOULD also set the submission properties, as specified in [MS-OXOMSG] section 2.2.3.

2.2.1   Namespaces

This specification defines and references various **XML namespaces** 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
evm	http://schemas.microsoft.com/exchange/um/2010/evm	

2.2.2   Voice Message

Voice messages and fax messages are **Message objects** that follow specific conventions, including:

- The value of the **PidTagMessageClass** property ([MS-OXOMSG] section 2.2.2.1) on the Message object, as specified in section 2.2.2.1.

- The format and order of voice message and fax attachments, as specified in section 2.2.2.2 and section 2.2.2.3.

- The use by the client of the **PidNameAudioNotes** property (section 2.2.5.15) for storing user annotations, as specified in section 2.2.2.4.

- The optional inclusion of speech-to-text data in the **PidNameAutomaticSpeechRecognitionData** property (section 2.2.5.13), as specified in section 2.2.2.5.

2.2.2.1   Message Classes

For voice messages, the value of the **PidTagMessageClass** property ([MS-OXOMSG] section 2.2.2.1) MUST be one the following:

- **IPM.Note.Microsoft.Voicemail.UM.CA** for original messages taken with audio content by telephone.

- **IPM.Note.Microsoft.Voicemail.UM** for original messages taken with audio content by telephone but not as a result of call answering (for example, if the phone of the **recipient (1)** did not ring).

- The value of the original **PidTagMessageClass** property suffixed with **.Microsoft.Voicemail** for messages with audio content that was created in response to other messages. For example, a voice reply to a message of type **IPM.Note** has the type **IPM.Note.Microsoft.Voicemail**.

For fax messages, the value of the **PidTagMessageClass** property MUST be set to **IPM.Note.Microsoft.FAX.CA**.

For missed call notifications, the value of the **PidTagMessageClass** property MUST be set to **IPM.Note.Microsoft.Missed.Voice**.

2.2.2.2   Attachments

Messages with audio content carry the audio content as a file attachment on the message, in accordance with the procedures for attachment handing as specified in [MS-OXCMSG] section 3.1.4.12. The attachment file MUST be in either the WAV file format (as specified in [WAVE]), the ASF file format (as specified in [ASF]), or the MP3 file format<1>.

If in the WAV format, the audio **codec** MUST be either G.711 a-law, G.711 m-law, or GSM 6.10, as specified in [G711] and [GSM610]. If in the ASF file format, the codec MUST be either the Windows Media Audio 9 Voice or the WMA 2 codec.

In addition to the common properties on the attachment, the attachment MUST define the following two properties:

- **PidTagAttachLongFilename** ([MS-OXCMSG] section 2.2.2.10): Set to a unique name in the attachment collection of the message. To function properly, the file name MUST be unique for the attachment order logic specified in section 2.2.2.3. The file extension MUST be ".wav" for files in the WAV format, MUST be ".wma" for files in the ASF format, and MUST be ".mp3" for files in the MP3 format.

- **PidTagAttachMimeTag** ([MS-OXCMSG] section 2.2.2.26): Set to reflect the audio content type of the message. The value of the property depends upon how the message is encoded:

  - For WMA 9 Voice-encoded messages, this value MUST be "audio/wma".

  - For GSM 6.10-encoded messages, this value MUST be "audio/gsm".

  - For G.711-encoded messages, this value MUST be "audio/WAV".

  - For MP3-encoded messages, this value MUST be "audio/mp3".

2.2.2.3   Attachment Order

Any message that contains audio attachments MUST define the **PidTagVoiceMessageAttachmentOrder** property (section 2.2.5.9).

2.2.2.4   Audio Notes

The client can enable a user to annotate a voice message with textual information after it has been delivered to the user's mailbox. For example, a user can note a telephone number or name that was included in the audio content of the message.

If the client saves that textual information on the message, it MUST set the **PidNameAudioNotes** property (section 2.2.5.15) to the value of that textual information.

2.2.2.5 ASR Data

Automatic speech recognition (ASR) data<2> refers to the text transcription of an audio attachment. In an unprotected voice message, this data is stored in the **PidNameAutomaticSpeechRecognitionData** property (section 2.2.5.13). In a protected voice message, it is handled as an attachment instead. As with other attachments in a **rights-managed e-mail message**, the attachment is stored in the Attachment List storage of the encrypted **binary large object (BLOB)**, as specified in [MS-OXORMMS] section 3.1.4.1.1.

A client or server can submit a voice message to a third party transcription service in order to obtain a translation of the original message in the ASR data format. The transmission of data to and from this third party service is outside the scope of this specification.

2.2.2.5.1 ASR XML Schema Definition

The ASR **XML schema** defines a format for storing ASR messages. The ASR XML conforms to the following XML schema.

```
<?xml version="1.0" encoding="UTF-8"?>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:evm="http://schemas.microsoft.com/exchange/um/2010/evm"
targetNamespace="http://schemas.microsoft.com/exchange/um/2010/evm">
    <!--Type definitions-->
    <xs:simpleType name="zeroToUnityDoubleType">
        <xs:restriction base="xs:double">
            <xs:minInclusive value="0.0"/>
            <xs:maxInclusive value="1.0"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="versionNumberType">
        <xs:restriction base="xs:token">
            <xs:pattern value="\d+\.\d+\.\d+\.\d+"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="recoObjectType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="ts" use="required">
                    <xs:simpleType>
                        <xs:restriction base="xs:time"/>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="te" use="required">
                    <xs:simpleType>
                        <xs:restriction base="xs:time"/>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="c" use="required">
                    <xs:simpleType>
                        <xs:restriction base="evm:zeroToUnityDoubleType"/>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="nx" use="optional">
                    <xs:simpleType>
                        <xs:restriction base="xs:token"/>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="id" use="required">
```

```
                <xs:simpleType>
                    <xs:restriction base="xs:ID"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="be" use="optional" default="1">
                <xs:simpleType>
                    <xs:restriction base="xs:boolean"/>
                </xs:simpleType>
            </xs:attribute>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="recoResultType">
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="skipped"/>
        <xs:enumeration value="attempted"/>
        <xs:enumeration value="partial"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="recoErrorType">
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="success"/>
        <xs:enumeration value="audioqualitypoor"/>
        <xs:enumeration value="languagenotsupported"/>
        <xs:enumeration value="rejected"/>
        <xs:enumeration value="badrequest"/>
        <xs:enumeration value="systemerror"/>
        <xs:enumeration value="timeout"/>
        <xs:enumeration value="messagetoolong"/>
        <xs:enumeration value="protectedvoicemail"/>
        <xs:enumeration value="throttled"/>
        <xs:enumeration value="other"/>
        <xs:enumeration value="errorreadingsettings"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="confidenceBandType">
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="low"/>
        <xs:enumeration value="medium"/>
        <xs:enumeration value="high"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="breakWeightType">
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="low"/>
        <xs:enumeration value="medium"/>
        <xs:enumeration value="high"/>
    </xs:restriction>
</xs:simpleType>
<!--Element definitions-->
<xs:element name="Break">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="evm:recoObjectType">
                <xs:attribute name="wt" type="evm:breakWeightType"
                    use="optional" default="medium"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
```

```
            </xs:element>
            <xs:element name="Text">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="evm:recoObjectType"/>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
            <xs:element name="Feature">
                <xs:complexType>
                    <xs:sequence maxOccurs="unbounded">
                        <xs:element ref="evm:Text"/>
                    </xs:sequence>
                    <xs:attribute name="class" use="required">
                        <xs:simpleType>
                            <xs:restriction base="xs:token"/>
                        </xs:simpleType>
                    </xs:attribute>
                    <xs:attribute name="reference" type="xs:token" use="optional"/>
                    <xs:attribute name="reference2" type="xs:token" use="optional"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="ErrorInformation">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:token">
                            <xs:attribute name="lang" type="xs:language" use="required"/>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
            <xs:element name="Information">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:string">
                            <xs:attribute name="lang" type="xs:language" use="required"/>
                            <xs:attribute name="linkURL" type="xs:anyURI" use="optional" default=""/>
                            <xs:attribute name="linkText" type="xs:normalizedString"
                                use="optional" default=""/>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
            <xs:element name="ASR">
                <xs:complexType>
                    <xs:choice minOccurs="0" maxOccurs="unbounded">
                        <xs:element ref="evm:ErrorInformation" minOccurs="0" maxOccurs="1"/>
                        <xs:element ref="evm:Text"/>
                        <xs:element ref="evm:Break"/>
                        <xs:element ref="evm:Feature"/>
                        <xs:element ref="evm:Information" minOccurs="0" maxOccurs="1"/>
                    </xs:choice>
                    <xs:attribute name="productID" type="xs:unsignedInt" use="optional" default="0"/>
                    <xs:attribute name="confidence" type="evm:zeroToUnityDoubleType" use="required"/>
                    <xs:attribute name="confidenceBand" type="evm:confidenceBandType"
                        use="optional" default="medium"/>
                    <xs:attribute name="lang" type="xs:language" use="required"/>
                    <xs:attribute name="productVersion" type="evm:versionNumberType" use="optional"/>
                    <xs:attribute name="recognitionError" type="evm:recoErrorType" use="required"/>
```

```
                <xs:attribute name="recognitionResult" type="evm:recoResultType" use="required"/>
                <xs:attribute name="schemaVersion" type="evm:versionNumberType" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

2.2.2.5.1.1   Simple Types

2.2.2.5.1.1.1   evm:breakWeightType Simple Type

The **breakWeightType** simple type represents a coarse classification of the magnitude of a break in the speech data that was processed to obtain a transcript.

```
<xs:simpleType name="breakWeightType">
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="low"/>
        <xs:enumeration value="medium"/>
        <xs:enumeration value="high"/>
    </xs:restriction>
</xs:simpleType>
```

The enumerated values for the **breakWeightType** simple type are defined as follows.

Value	Meaning
low	A low break weight was used.
medium	A medium break weight was used.
high	A high break weight was used.

2.2.2.5.1.1.2   evm:confidenceBandType Simple Type

The **confidenceBandType** simple type represents a coarse classification of a confidence result (that is itself represented as an **zeroToUnityDoubleType** simple type). A value of "low" indicates that the transcript is probably significantly inaccurate. The heuristics for classification are not described here.

```
<xs:simpleType name="confidenceBandType">
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="low"/>
        <xs:enumeration value="medium"/>
        <xs:enumeration value="high"/>
    </xs:restriction>
</xs:simpleType>
```

The enumerated values for the **confidenceBandType** simple type are defined as follows.

Value	Meaning
low	The transcription is of low (possibly poor) quality.
medium	The transcription is of average quality.

Value	Meaning
high	The transcription is of high quality.

2.2.2.5.1.1.3 evm:recoErrorType Simple Type

The **recoErrorType** simple type represents success or the types of errors returned by the voice message transcription service.

```
<xs:simpleType name="recoErrorType">
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="success"/>
        <xs:enumeration value="audioQualityPoor"/>
        <xs:enumeration value="languageNotSupported"/>
        <xs:enumeration value="rejected"/>
        <xs:enumeration value="badRequest"/>
        <xs:enumeration value="systemError"/>
        <xs:enumeration value="timeout"/>
        <xs:enumeration value="messagetoolong"/>
        <xs:enumeration value="protectedvoicemail"/>
        <xs:enumeration value="throttled"/>
        <xs:enumeration value="errorreadingsettings"/>
        <xs:enumeration value="other"/>
    </xs:restriction>
</xs:simpleType>
```

The enumerated values for the **recoErrorType** simple type are defined as follows.

Value	Meaning
success	The transcription was successfully completed.
audioQualityPoor	The quality of the recording was too low to complete a transcript. This can be caused by low volume, high noise, distortion, sound drop-out, or some combination of all of these elements.
languageNotSupported	The transcription service cannot process the spoken language used in the voice message.
rejected	The voice message audio does not conform to the requirements of the transcription system.
badRequest	The voice message request to the transcription service was not well formed.
systemError	An unexpected error prevented transcription.
timeout	The voice transcription process took too long and was stopped.
messagetoolong	The voice message was too lengthy to be transcribed.
protectedvoicemail	The voice message has rights protection enabled, and cannot be transcribed.
throttled	Bandwidth or network limitations prevent this voice message from being transcribed.
errorreadingsettings	The transcription service cannot read the transcription settings of the user's mailbox.

Value	Meaning
other	An unknown error occurred during voice transcription.

2.2.2.5.1.1.4 evm:recoResultType Simple Type

The **recoResultType** simple type represents the result types for voice recognition.

```
<xs:simpleType name="recoResultType">
    <xs:restriction base="xs:NCName">
        <xs:enumeration value="skipped"/>
        <xs:enumeration value="attempted"/>
        <xs:enumeration value="partial"/>
    </xs:restriction>
</xs:simpleType>
```

The enumerated values for the **recoResultType** simple type are defined as follows.

Value	Meaning
skipped	The transcription service did not attempt to translate the voice message.
attempted	The transcription service tried to translate the voice message.
partial	The transcription service provided an incomplete transcription of the voice message.

2.2.2.5.1.1.5 evm:versionNumberType Simple Type

The **evm:versionNumberType** simple type represents the server version number format.

```
<xs:simpleType name="versionNumberType">
    <xs:restriction base="xs:token">
        <xs:pattern value="\d+\.\d+\.\d+\.\d+"/>
    </xs:restriction>
</xs:simpleType>
```

2.2.2.5.1.1.6 evm:zeroToUnityDoubleType

The **evm:zeroToUnityDoubleType** simple type represents probabilistic information.

```
<xs:simpleType name="zeroToUnityDoubleType">
    <xs:restriction base="xs:double">
        <xs:minInclusive value="0.0"/>
        <xs:maxInclusive value="1.0"/>
    </xs:restriction>
</xs:simpleType>
```

2.2.2.5.1.2 Complex Types

2.2.2.5.1.2.1 evm:recoObjectType Complex Type

The **evm:recoObjectType** complex type represents information for a section of a voice recognition transcript.

```
<xs:complexType name="recoObjectType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="ts" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:time"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="te" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:time"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="c" use="required">
                <xs:simpleType>
                    <xs:restriction base="evm:zeroToUnityDoubleType"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="nx" use="optional">
                <xs:simpleType>
                    <xs:restriction base="xs:token"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="id" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:ID"/>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="be" use="optional" default="1">
                <xs:simpleType>
                    <xs:restriction base="xs:boolean"/>
                </xs:simpleType>
            </xs:attribute>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

The attributes of the **evm:recoObjectType** complex type are specified as follows. Any data types not specified in this document are specified in [XMLSCHEMA2/2].

Attribute	Type	Definition
be	**xs:Boolean**	Optional. Indicates whether the element is calculated to be on the most probable (1-best) path through the transcript (if "1" or "true"), or not (if "0" or "false").
c	**evm:zeroToUnityDoubleType** (section 2.2.2.5.1.1.6)	Required. Indicates the speech recognition system's confidence in this suggestion.

Attribute	Type	Definition
id	**xs:ID**	Required. Uniquely identifies the element within the transcript.
nx	**xs:token**	Optional. If this is not the final element of the transcript, the value of the attribute contains the identifier (ID) of the following element—that is, the next in time order.
te	**xs:time**	Required. Indicates the time (measured from the start of the audio) at which the corresponding message ends.
ts	**xs:time**	Required. Indicates the time (measured from the start of the audio) at which the corresponding message begins.

2.2.2.5.1.3   Elements

2.2.2.5.1.3.1   ASR Element

The **ASR** element is the root element of a transcript. Its attributes refer to the transcript as a whole. It contains elements that describe individual recognition objects (words, numbers, pauses, and so on) and possibly also describe associated features (names, telephone numbers, and so on).

```
<xs:element name="ASR">
    <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="evm:ErrorInformation" minOccurs="0"/>
            <xs:element ref="evm:Text"/>
            <xs:element ref="evm:Break"/>
            <xs:element ref="evm:Feature"/>
        </xs:choice>
    <xs:attribute name="productID" type="xs:unsignedInt" use="optional" default="0"/>
    <xs:attribute name="confidence" type="evm:zeroToUnityDoubleType" use="required"/>
    <xs:attribute name="confidenceBand" type="evm:confidenceBandType"
        use="optional" default="medium"/>
    <xs:attribute name="lang" type="xs:language" use="required"/>
    <xs:attribute name="productVersion" type="evm:versionNumberType" use="optional"/>
    <xs:attribute name="recognitionError" type="evm:recoErrorType" use="optional"
        default="success"/>
    <xs:attribute name="recognitionResult" type="evm:recoResultType" use="required"/>
    <xs:attribute name="schemaVersion" type="evm:versionNumberType" use="required"/>
    </xs:complexType>
</xs:element>
```

The **ASR** element has the following attributes. Any data types not specified in this document are specified in [XMLSCHEMA2/2].

Attribute	Type	Definition
confidence	**evm:zeroToUnityDoubleType** (section 2.2.2.5.1.1.6)	Required. Indicates the overall confidence in the recognition results. This is calculated by the speech recognition system as a weighted average over the individual recognition

Attribute	Type	Definition
		elements.
confidenceBand	evm:confidenceBandType (section 2.2.2.5.1.1.2)	Optional. Provides a general indication of the system's overall confidence in the recognition results.
lang	xs:language	Required. Indicates the language in which the attempt at automatic speech recognition was made.
productID	xs:unsignedInt	Optional. If present, this attribute identifies the product or service that was used to produce the transcript. Values will be assigned to partner products and services by Microsoft. Partners MUST<3> provide their ID when sending the transcript.
productVersion	evm:versionNumberType (section 2.2.2.5.1.1.5)	Optional. If present, indicates the version of the software that was used to produce the transcript.<4>
recognitionError	evm:recoErrorType (section 2.2.2.5.1.1.3)	Optional. If present, provides for a more specific indication of the success or failure of the recognition than does the **recognitionResult** attribute.
recognitionResult	evm:recoResultType (section 2.2.2.5.1.1.4)	Required. Indicates whether an attempt at recognition was made and, if so, whether the recognition was completed.
schemaVersion	evm:versionNumberType	Required. Indicates the version of the schema description. This SHOULD be "1.0.0.0".

2.2.2.5.1.3.2   Break Element

The **Break** element represents a discontinuity in the semantic content of a recording. For example, the speech might have paused for significantly longer than the typical amount of time between words. There is no expected value; all relevant information is contained in the attributes.

```
<xs:element name="Break">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="evm:recoObjectType">
                <xs:attribute name="wt" type="evm:breakWeightType" use="optional"
default="medium"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
```

The **Break** element has the following attributes.

Attribute	Type	Definition
wt	evm:breakWeightType (section 2.2.2.5.1.1.1)	Optional. Indicates the magnitude of the break.

2.2.2.5.1.3.3   ErrorInformation

The **ErrorInformation** element provides a mechanism for the partner to return more detailed information when the **recognitionError** attribute of the **ASR** element, as specified in section 2.2.2.5.1.3.1, is set to a value other than "success". The content of the element is expected to contain some diagnostic information that can help recipients (1) of the document to understand why the transcript was not produced as expected. This element is required and expected only when the **recognitionResult** attribute of the **ASR** element has a value of either "skipped" or "partial". It can also be omitted unless the **recognitionError** attribute of the **ASR** element has a value of "other".

```
<xs:element name="ErrorInformation">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:token">
                <xs:attribute name="lang" type="xs:language" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
```

The **ErrorInformation** element has the following attributes. Any data types not specified in this document are specified in [XMLSCHEMA2/2].

Attribute	Type	Definition
lang	**xs:language**	Required. Indicates the language in which the error description is written. This is not required to be the same as the language in which the attempt at speech recognition was made.

2.2.2.5.1.3.4   Feature Element

The **Feature** element represents an assignment of special meaning to one or more **Text** elements in the transcript. The **Text** elements are contained within the **Feature** element. Any data types not specified in this document are specified in [XMLSCHEMA2/2].

```
<xs:element name="Feature">
    <xs:complexType>
        <xs:sequence maxOccurs="unbounded">
            <xs:element ref="evm:Text"/>
        </xs:sequence>
        <xs:attribute name="class" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:token"/>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="reference" type="xs:token" use="optional"/>
        <xs:attribute name="reference2" type="xs:token" use="optional"/>
    </xs:complexType>
</xs:element>
```

The **Feature** element has the following attributes. Any data types not specified in this document are specified in [XMLSCHEMA2/2].

Attribute	Type	Definition
class	**xs:token**	Required. Indicates the type of feature that has been identified.
reference	**xs:token**	Optional. If data relevant to the **Feature** markup exists outside the transcript, this attribute will contain a pointer that will enable an application to locate and (with sufficient permission) access the data.
reference2	**xs:token**	Optional. If data relevant to the **Feature** markup exists outside the transcript, this attribute will contain a pointer that will enable an application to locate and (with sufficient permission) access the data.

The supported values of the **class** attribute of the **Feature** element are listed in the following table.

Feature class name	Reference?	Description
Contact	Yes	A personal contact of the Unified Messaging-enabled user to whom the voice message was sent. The reference is the Item ID of the **Contact object**, as returned by the server.
Date	Yes	A date. The reference represents a canonical version of the date. This can be in either an **xs:date** format, as specified in [XMLSCHEMA2/2], or a regional format deduced from the recognition language that is being used.
Mailbox	Yes	A mailbox-enabled user. The reference is the primary **Simple Mail Transfer Protocol (SMTP)** address of the user.
PersonName	Yes	A person's name. The reference has the same value as the contained text.
PhoneNumber	No	A series of digits (and possibly other characters), probably representing a telephone number. The value can be expanded to a canonical form in line with regional conventions that are deduced from the recognition language that is being used.

2.2.2.5.1.3.5   Text

The **Text** element represents a portion of a transcript that can be a single word or number. This is contained as the value of the element.

```
<xs:element name="Text">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="evm:recoObjectType"/>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
```

2.2.2.5.1.3.6   Information

The **Information** element represents additional metadata regarding the transcript.

```
<xs:element name="Information">
```

```
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:string">
                        <xs:attribute name="lang" type="xs:language" use="required"/>
                        <xs:attribute name="linkURL" type="xs:anyURI" use="optional" default=""/>
                        <xs:attribute name="linkText" type="xs:normalizedString"
                            use="optional" default=""/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
```

The **Information** element has the following attributes. Any data types not specified in this document are specified in [XMLSCHEMA2/2].

Attribute	Type	Definition
lang	**xs:language**	Required. Indicates the language used for transcription.
linkURL	**xs:anyURI**	Optional. The **URL** where the transcript file can be obtained.
linkText	**xs:normalizedString**	Optional. The text for the **linkURL** attribute.

2.2.3   Protected Voice Message

A protected voice message<5> is similar to a rights-managed e-mail message, as specified in [MS-OXORMMS] section 2.2.1. However, the client application needs to be aware of subtle differences between a rights-managed e-mail message and a protected voice message when rendering protected voice messages.

2.2.3.1   Messages

2.2.3.1.1   Message Classes

A protected voice message is represented by the following **message classes**:

- **IPM.NOTE.rpmsg.Microsoft.VoiceMail.UM.CA**, for original messages taken with audio content by telephone as a result of call answering.

- **IPM.NOTE.rpmsg.Microsoft.VoiceMail.UM**, for original messages taken with audio content by telephone as a result of any scenario other than call answering.

2.2.3.1.2   Message Content

As specified in [MS-OXORMMS], a rights-managed e-mail message consists of a wrapper message with the original e-mail content encrypted as a BLOB in an attachment. The attachment has the following properties:

- **PidNameContentClass** ([MS-OXCMSG] section 2.2.1.48): MUST be set to "rpmsg.message".

- **PidTagAttachLongFilename** ([MS-OXCMSG] section 2.2.2.11): MUST be set to "message.rpmsg".

- **PidTagAttachMimeTag** ([MS-OXCMSG] section 2.2.2.26): MUST be set to "application/x-microsoft-rpmsg-message".

A protected voice message follows this convention. A nonprotected voice message contains one or more audio attachments and voice message preview data in the **PidNameAutomaticSpeechRecognitionData** property (section 2.2.5.13). In the case of a protected voice message, the audio attachment(s) and voice message preview data are treated as attachments and are stored within the encrypted BLOB. These attachments MUST be stored in the Attachment List storage, as specified in [MS-OXORMMS] section 2.2.4.4.2.

2.2.3.2 Audio Attachments

Audio attachments carry the audio content of a voice message.

When an audio attachment is added to the Attachment List storage in the encrypted BLOB, it is encrypted. Depending on the original codec that is used to encode the audio attachment, the encrypted audio attachment carries the file name extension "umrmwav", "umrmwma", or "umrmmp3".

The content of the **PidTagVoiceMessageAttachmentOrder** property (section 2.2.5.9) in an unprotected voice message contains the list of the file names of the audio attachments. This is true for protected voice messages, except that all of the attachment file names have the ".umrmwav", ".umrmwma", or ".umrmmp3" extension.

2.2.3.3 Protected Voice Message Property

The **PidNameXRequireProtectedPlayOnPhone** property (section 2.2.5.14) is set on the outer message of the protected voice message. When this property is set to "TRUE", the client that renders this message MUST NOT allow users to listen to the voice attachment by means of the e-mail client. The client MUST offer the Play-On-Phone feature to the user as the only option for listening to the voice message.

2.2.4 UI Configuration

A client application can display an enhanced user interface (UI) for Message objects with the message classes specified in section 2.2.2.1 for some users and not for others. In addition, the client can show UI configuration information related to a user's telephony experience for some users and not for others. The server SHOULD store settings for these options on a per-user basis, and the client MUST consult these settings before it attempts to implement the aforementioned UI segmentation.

This could be useful in a scenario in which a certain group of users are not provisioned by their administrator to receive the message classes specified in section 2.2.2.1 and/or are not provisioned to have telephony access to their messages.

If the client or server sets or uses this configuration information, it MUST treat this information as a **dictionary stream (1)** by using the Configuration Information Protocol, as specified in [MS-OXOCFG].

The dictionary stream (1) object MUST be stored in the Inbox **special folder**, as specified in [MS-OXOSFLD] section 2.2.7.

The dictionary stream (1) MUST have the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) set on it. The value of this property MUST be **IPM.Configuration.UMOLK.UserOptions**.

The dictionary stream (1) SHOULD include the following *outlookFlags* parameter. If the *outlookFlags* parameter does not appear in the dictionary stream (1) or the dictionary stream (1) does not exist, the default value 0x00000000 SHOULD be assumed.

- Name (string): "outlookFlags"

- Value (32-bit integer): The least significant bit MUST correspond to whether the client displays special UI information for message classes that are specified in section 2.2.2.1. The second-least significant bit MUST correspond to whether the client displays telephony configuration UI to the user. The four possible values are listed in the following table; the value 0x00000000 is the default.

Value	Meaning
0x00000000	Display neither the special UI information for message classes nor the telephony configuration UI.
0x00000001	Display only the special UI information for message classes.
0x00000002	Display only the telephony configuration UI.
0x00000003	Display both the special UI information for message classes and the telephony configuration UI.

2.2.5 Message Object Properties

Message object properties that can be defined on Message objects that contain voice messages and protected voice messages are specified in section 2.2.5.1 through section 2.2.5.15. Message objects are further specified in [MS-OXCMSG].

2.2.5.1 PidTagSenderTelephoneNumber Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidTagSenderTelephoneNumber** property ([MS-OXPROPS] section 2.996) contains the telephone number of the caller associated with a voice message.

The relationship between this property and the **X-CallingTelephoneNumber MIME header (2)** is specified in [MS-OXCMAIL] section 2.1.3.2.3.

2.2.5.2 PidNameXSenderTelephoneNumber Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidNameXSenderTelephoneNumber** property ([MS-OXPROPS] section 2.477) contains the telephone number of the caller associated with a voice message.

The relationship between this property and the **X-CallingTelephoneNumber** MIME header (2) is specified in [MS-OXCMAIL] section 2.1.3.2.3 and [MS-OXCMAIL] section 2.2.3.2.23.

2.2.5.3 PidTagVoiceMessageDuration Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagVoiceMessageDuration** property ([MS-OXPROPS] section 2.1048) specifies the length of the attached voice message, in seconds.

The relationship between this property and the **X-VoiceMessageDuration** MIME header (2) is specified in [MS-OXCMAIL] section 2.1.3.2.3 and [MS-OXCMAIL] section 2.2.3.2.23.

2.2.5.4 PidNameXVoiceMessageDuration Property

Type: **PtypInteger16** ([MS-OXCDATA] section 2.11.1)

The **PidNameXVoiceMessageDuration** property ([MS-OXPROPS] section 2.494) specifies the length of the attached voice message, in seconds.

The relationship between this property and the **X-VoiceMessageDuration** MIME header (2) is specified in [MS-OXCMAIL] section 2.1.3.2.3 and [MS-OXCMAIL] section 2.2.3.2.23.

2.2.5.5 PidTagVoiceMessageSenderName Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidTagVoiceMessageSenderName** property ([MS-OXPROPS] section 2.1049) specifies the name of the caller who left the attached voice message, as provided by the voice network's caller ID system.

The relationship between this property and the **X-VoiceMessageSenderName** MIME header (2) is specified in [MS-OXCMAIL] section 2.1.3.2.3 and [MS-OXCMAIL] section 2.2.3.2.23.

2.2.5.6 PidNameXVoiceMessageSenderName Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1.2)

The **PidNameXVoiceMessageSenderName** property ([MS-OXPROPS] section 2.495) specifies the name of the caller who left the attached voice message, as provided by the voice network's caller ID system.

The relationship between this property and the **X-VoiceMessageSenderName** MIME header (2) is specified in [MS-OXCMAIL] section 2.1.3.2.3 and [MS-OXCMAIL] section 2.2.3.2.23.

2.2.5.7 PidTagFaxNumberOfPages Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagFaxNumberOfPages** property ([MS-OXPROPS] section 2.686) specifies how many discrete pages are contained within an attachment representing a facsimile message.

The relationship between this property and the **X-FaxNumberOfPages** MIME header (2) is specified in [MS-OXCMAIL] section 2.1.3.2.3 and [MS-OXCMAIL] section 2.2.3.2.23.

2.2.5.8 PidNameXFaxNumberOfPages Property

Type: **PtypInteger16** ([MS-OXCDATA] section 2.11.1)

The **PidNameXFaxNumberOfPages** property ([MS-OXPROPS] section 2.475) specifies how many discrete pages are contained within an attachment representing a facsimile message.

The relationship between this property and the **X-FaxNumberOfPages** MIME header (2) is specified in [MS-OXCMAIL] section 2.1.3.2.3 and [MS-OXCMAIL] section 2.2.3.2.23.

2.2.5.9 PidTagVoiceMessageAttachmentOrder Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagVoiceMessageAttachmentOrder** property ([MS-OXPROPS] section 2.1047) contains the list of names for the audio file attachments that are to be played as part of a message, in reverse order. The file names are separated by semicolons.

The content of this property is a list of values assigned to the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.10) for audio file attachments that are to be played as part of the message. The order MUST be the reverse of the order in which the attachments were added; that is, the most recently added message first, the next most recently added message second, and so on.

The file names MUST be separated by semicolons. Each file name can be prefixed or suffixed with spaces. The first file name in the list can be preceded by a semicolon, and the last file name in the list can be suffixed with a semicolon.

For example, for a message that contains only one voice file attachment, for which the value of the **PidTagAttachLongFilename** property is "vm.wav", acceptable values for the **PidTagVoiceMessageAttachmentOrder** property include but are not limited to the following:

- vm.wav

- ;vm.wav

- ; vm.wav

Or, for example, a message contains three attachments, for which the **PidTagAttachLongFilename** property values are "vm1.wav", "vm2.wav", and "vm3.wav". The files were added in the order "vm1.wav", then "vm2.wav", and then "vm3.wav". Acceptable values for the **PidTagVoiceMessageAttachmentOrder** property include but are not limited to the following:

- vm3.wav;vm2.wav;vm1.wav

- vm3.wav; vm2.wav; vm1.wav

- ;vm3.wav;vm2.wav;vm1.wav

- Vm3.wav;vm2.wav;vm1.wav

- ;vm3.wav;vm2.wav;vm1.wav

The relationship between this property and the **X-AttachmentOrder** MIME header (2) is specified in [MS-OXCMAIL] section 2.1.3.2.3 and [MS-OXCMAIL] section 2.2.3.2.23.

2.2.5.10   PidNameXVoiceMessageAttachmentOrder Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidNameXVoiceMessageAttachmentOrder** property ([MS-OXPROPS] section 2.493) contains the list of names for the audio file attachments that are to be played as part of a voice message, in reverse order. The file names are separated by semicolons.

The format of this property is identical to the format of the **PidTagVoiceMessageAttachmentOrder** property (section 2.2.5.9).

The relationship between this property and the **X-AttachmentOrder** MIME header (2) is specified in [MS-OXCMAIL] section 2.1.3.2.3 and [MS-OXCMAIL] section 2.2.3.2.23.

2.2.5.11   PidTagCallId Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagCallId** property ([MS-OXPROPS] section 2.619) is a unique identifier associated with the phone call.

The relationship between this property and the MIME header (2) is specified in [MS-OXCMAIL] section 2.1.3.2.3 and [MS-OXCMAIL] section 2.2.3.2.23.

2.2.5.12   PidNameXCallId

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidNameXCallId** property ([MS-OXPROPS] section 2.474) is a unique identifier associated with the phone call.

The relationship between this property and the MIME header (2) is specified in [MS-OXCMAIL] section 2.1.3.2.3 and [MS-OXCMAIL] section 2.2.3.2.23.

2.2.5.13   PidNameAutomaticSpeechRecognitionData Property

Type: **PtypBinary** ([MS-OXCDATA] section 2.11.1)

The **PidNameAutomaticSpeechRecognitionData** property ([MS-OXPROPS] section 2.372) contains the automated text transcription of the attached voice message.

Further details on the format of this property are specified in section 2.2.2.5.

2.2.5.14   PidNameXRequireProtectedPlayOnPhone Property

Type: **PtypBoolean** ([MS-OXCDATA] section 2.11.1)

The **PidNameXRequireProtectedPlayOnPhone** property ([MS-OXPROPS] section 2.476) specifies whether a protected voice message can only be played over the phone.

Further details on the format of this property are specified in section 2.2.3.3.

2.2.5.15   PidNameAudioNotes Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidNameAudioNotes** property ([MS-OXPROPS] section 2.370) is an optional property set by the client that contains any notes added by the user to the voice message.

3   Protocol Details

3.1   Client Details

The client role is to display the Unified Messaging objects specified in section 2.2.2.1. There are two possible levels of client experience: down-level and up-level.

A "down-level" experience does nothing apart from the basic client role specified in [MS-OXCMSG] for Message objects. For an example of this experience, see section 4.1.1.

Alternatively, the client can provide an "up-level" experience for displaying Unified Messaging objects, including the ability to edit audio notes (section 2.2.2.4) and/or providing a means to automatically play back the audio content of a message by using the attachments (section 2.2.2.2) and the attachment order information (section 2.2.2.3). For an example of this experience, see section 4.1.2.

3.1.1   Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The client-side abstract data model for this protocol is specified in [MS-OXOMSG].

3.1.2   Timers

None.

3.1.3   Initialization

None.

3.1.4   Higher-Layer Triggered Events

3.1.4.1   Playing an Audio Message That Has Multiple Attachments

To play a voice message that has multiple attachments, a client SHOULD consult the **PidTagVoiceMessageAttachmentOrder** property (section 2.2.5.9) to determine the proper playback order.

3.1.5   Message Processing Events and Sequencing Rules

None.

3.1.6   Timer Events

None.

3.1.7   Other Local Events

None.

3.2   Server Details

The server role in this protocol is to create the message types, as specified in section 2, in addition to the core server behavior as specified in [MS-OXCMSG].

When the server receives a message of one of the types specified in this document, the following additional properties MAY be set:

- **PidTagVoiceMessageSenderName** property (section 2.2.5.5)

- **PidTagSenderTelephoneNumber** property (section 2.2.5.1)

- **PidTagVoiceMessageDuration** property (section 2.2.5.3)

- **PidTagCallId** property (section 2.2.5.11)

3.2.1   Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The server-side abstract data model for this protocol is specified in [MS-OXOMSG].

3.2.2   Timers

None.

3.2.3   Initialization

None.

3.2.4   Higher-Layer Triggered Events

3.2.4.1   Creating a Voice Message

To create a voice message, the server MUST set the appropriate value of the **PidTagMessageClass** property ([MS-OXOMSG] section 2.2.2.1) as specified in section 2.2.2.1.

The server MUST add the audio content for a voice message as a file attachment on the message, in accordance with the procedures for attachment handling, as specified in [MS-OXCMSG] section 3.1.4.12. The server MUST set the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.10) and the **PidTagAttachMimeTag** property ([MS-OXCMSG] section 2.2.2.26) as specified in section 2.2.2.2.

In some situations, a client or server can add more than one audio attachment to a particular message. For example, a voice reply to a voice message can include the original voice content for reference. In such situations, the server SHOULD add an attachment for each voice segment and define the order using the **PidTagVoiceMessageAttachmentOrder** property (section 2.2.5.9).

3.2.5   Message Processing Events and Sequencing Rules

None.

3.2.6   Timer Events

None.

3.2.7   Other Local Events

None.

4   Protocol Examples

4.1   Playing a Voice Message

The examples in section 4.1.1 and section 4.1.2 both assume that a voice message has been stored by the server, as specified in section 2.

4.1.1   Down-Level Experience

A client consults the configuration information specified in section 2.2.4 and sees that the *outlookFlags* parameter setting indicates that the client provides a down-level experience for the voice message object that it is about to display.

To provide the down-level experience, the client renders the voice message with all the functionality it would give to a typical Message object, as described in [MS-OXOMSG]. In particular, it enables the user to access the audio attachment that is included in the message by using the standard mechanism provided by the client for accessing attachments.

Having accessed the content of the audio attachment, the user uses an audio player application on his or her local computer that supports the attachment codec to play the audio content.

4.1.2   Up-Level Experience

A client consults the configuration information specified in section 2.2.4 and sees that the *outlookFlags* parameter setting indicates that the client provides an up-level experience.

The up-level experience of the client includes the ability to click a single "Play" button and hear all audio attachments on the message played in the reverse order in which the attachments were added. The user clicks this button, and the client consults the attachment order information on the message (section 2.2.2.3) and sees that the value is "vm2.wma;vm1.wma". From this value, the client knows that there are two attachments on the voice message object with the **PidTagAttachLongFilename** property ([MS-OXCMSG] section 2.2.2.10) values "vm2.wma" and "vm1.wma", respectively.

The client downloads the attachment named "vm2.wma" and uses an audio player on the user's local computer to play the WMA 9 Voice audio content; it recognizes that the attachment is encoded with WMA 9 Voice because the **PidTagAttachMimeTag** property ([MS-OXCMSG] section 2.2.2.26) value of the attachment is "audio/wma". After the audio finishes playing, the client downloads "vm1.wma" and plays it in the same way.

The client up-level experience of the client application also includes the ability to read and edit audio notes directly on the voice message, and the user uses this feature. The client provides an editable area on the screen into which the user can type text. When the user is finished, the client persists the text in the **PidNameAudioNotes** property (section 2.2.5.15) of the voice message object. The next time the user views this particular voice message object, he sees the notes he typed because the client displays the content of the **PidNameAudioNotes** property of the voice message object.

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations that are specific to the Voice Mail and Fax Objects Protocol. Note, however, that general security considerations that pertain to the underlying transport do apply to this protocol. For more information, see [MS-OXCMSG].

5.2 Index of Security Parameters

None.

6 Appendix A: 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 2003

- Microsoft Exchange Server 2007

- Microsoft Exchange Server 2010

- Microsoft Exchange Server 2013

- Microsoft Office Outlook 2003

- Microsoft Office Outlook 2007

- Microsoft Outlook 2010

- Microsoft Outlook 2013

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

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

<1> Section 2.2.2.2: Exchange 2003 and Exchange 2007 do not support the MP3 format.

<2> Section 2.2.2.5: ASR data is not available in Exchange 2003 and Exchange 2007.

<3> Section 2.2.2.5.1.3.1: Exchange 2010 and Exchange 2013 insert a value of "925712" in transcripts that it generates.

<4> Section 2.2.2.5.1.3.1: Transcripts that are generated by Unified Messaging in Exchange 2010 and Exchange 2013 take the form "14.nn.nnnn.nnn", with n representing digits.

<5> Section 2.2.3: Protected voice mail is not available in Exchange 2003 and Exchange 2007.

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index