

[MS-OXOUM]: Voice Mail and Fax Objects 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.
- **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			
Author	Date	Version	Comments
Microsoft Corporation	April 4, 2008	0.1	Initial Availability.
Microsoft Corporation	April 25, 2008	0.2	Revised and updated property names and other technical content.
Microsoft Corporation	June 27, 2008	1.0	Initial Release.
Microsoft Corporation	August 6, 2008	1.01	Updated references to reflect date of initial release.
Microsoft Corporation	September 3, 2008	1.02	Revised and edited technical content.
Microsoft Corporation	December 3, 2008	1.03	Minor editorial fixes.
Microsoft Corporation	March 4, 2009	1.04	Revised and edited technical content.
Microsoft Corporation	April 10, 2009	2.0	Updated technical content and applicable product releases.

Table of Contents

1	Introduction.....	5
1.1	Glossary.....	5
1.2	References.....	6
1.2.1	Normative References.....	6
1.2.2	Informative References.....	7
1.3	Protocol Overview.....	7
1.4	Relationship to Other Protocols.....	7
1.5	Prerequisites/Preconditions.....	7
1.6	Applicability Statement.....	7
1.7	Versioning and Capability Negotiation.....	7
1.8	Vendor-Extensible Fields.....	8
1.9	Standards Assignments.....	8
2	Messages.....	8
2.1	Transport.....	8
2.1.1	Inherited Properties.....	8
2.1.2	Message-Specific Properties.....	8
2.1.2.1	Message Classes.....	8
2.1.2.2	Attachments.....	9
2.1.2.3	Attachment Order.....	9
2.1.2.4	Audio Notes.....	10
2.1.2.5	ASR Data.....	10
2.1.2.5.1	ASR XML.....	11
2.2	Protected Voice Message.....	19
2.2.1	Messages.....	19
2.2.1.1	Message Classes.....	19
2.2.1.2	Message Content.....	19
2.2.2	Audio Attachments.....	19
2.2.2.1	Encrypted Audio Attachment.....	20
2.2.3	Protected Voice Message Property.....	20
2.3	UI Configuration.....	20
3	Protocol Details.....	21
3.1	Client Details.....	21
3.1.1	Abstract Data Model.....	21
3.1.2	Timers.....	22
3.1.3	Initialization.....	22
3.1.4	Higher-Layer Triggered Events.....	22
3.1.5	Message Processing Events and Sequencing Rules.....	22
3.1.6	Timer Events.....	22
3.1.7	Other Local Events.....	22
3.2	Server Details.....	22

4	<i>Protocol Examples</i>	22
4.1	Playing a Voice Message	22
4.1.1	Down-Level Experience	22
4.1.2	Up-Level Experience	23
5	<i>Security</i>	23
5.1	Security Considerations for Implementers	23
5.2	Index of Security Parameters	23
6	<i>Appendix A: Office/Exchange Behavior</i>	23
	<i>Index</i>	25

1 Introduction

The Voice Mail and Fax Objects protocol provides the methods that are used by the server to manipulate Unified Messaging objects. This document specifies the properties on the Unified Messaging objects. This document does not specify **message** properties, which are defined in [MS-OXOMSG].

1.1 Glossary

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

- attachment**
- binary large object (BLOB)**
- folder**
- message**
- Message object**
- property**
- recipient**
- Simple Mail Transfer Protocol (SMTP)**
- special folder**

The following terms are specific to this document:

audio notes: Textual notes that can be attached to a **voice message**.

codec: Software that compresses and decompresses audio and video data streams.

fax message: A message stored in the server for a user that contains image content received from a calling fax machine; a digital representation of a typical fax message.

G.711: An ITU-T standard for audio compression that is typically used in telephony systems. There are two different implementations: a-law and m-law.

GSM 6.10: A form of audio compression that is used by most European wireless telephone systems.

missed call notification: A message stored on the server for a user that is intended to convey information about a call that was missed. The message contains information about the calling party, if available, and the time of call, but does not contain audio content because the calling party chose not to record a message.

rights-managed e-mail message: An e-mail **message** that specifies permissions that are designed to protect its content from inappropriate access, use, and distribution.

voice message: A message stored on the server for a user that contains audio content recorded by a calling party.

WAV: The main format for storing audio on Window's computers.

WMA 9 Voice: A form of audio compression released by Microsoft as part of the Windows Media Audio 9 SDK tool kit.

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

[G711] ITU-T, "Pulse Code Modulation (PCM) of Voice Frequencies", Recommendation G.711, June 1990, <http://www.itu.int/rec/T-REC-G.711/e>.

[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-DTYP] Microsoft Corporation, "Windows Data Types", March 2007, <http://go.microsoft.com/fwlink/?LinkId=111558>.

[MS-OXCMSG] Microsoft Corporation, "Message and Attachment Object Protocol Specification", June 2008.

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

[MS-OXOCFG] Microsoft Corporation, "Configuration Information Protocol Specification", June 2008.

[MS-OXOMSG] Microsoft Corporation, "E-Mail Object Protocol Specification", June 2008.

[MS-OXORMMS] Microsoft Corporation, "Rights-Managed E-Mail Object Protocol Specification", June 2008.

[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol Specification", June 2008.

[MS-OXPROPS] Microsoft Corporation, "Exchange Server Protocols Master Property List Specification", June 2008.

[MS-OXWUMS] Microsoft Corporation, "Voice Mail Settings Web Service Protocol Specification", June 2008.

[MSDN-ASF] Microsoft Corporation, "Advanced Systems Format (ASF) Specification", <http://go.microsoft.com/fwlink/?LinkId=114334>.

[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>.

[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>.

1.2.2 Informative References

[MSDN-WMSSDK] Microsoft Corporation, "Windows Media Services 9.0 Series SDK", <http://go.microsoft.com/fwlink/?LinkId=114332>.

1.3 Protocol Overview

Unified Messaging objects are items created on behalf of telephone callers or fax senders by the server and 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**, as defined in section 1.1.

1.4 Relationship to Other Protocols

The Voice Mail and Fax Objects protocol specification relies on an understanding of how to work with **properties**, **folders**, **messages**, and **attachments** (for more details, see [MS-OXPROPS], [MS-OXOSFLD], and [MS-OXCMSG]).

The Voice Mail and Fax Objects protocol uses the Message and Attachment Object protocol, as specified in [MS-OXCMSG], as a transport protocol between the client and the server.

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 which is already specified in [MS-OXCMSG].

1.9 Standards Assignments

None.

2 Messages

There are three types of Unified Messaging objects: **voice messages**, **missed call notifications**, and **fax 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.

2.1.1 Inherited Properties

None.

2.1.2 Message-Specific Properties

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

2.1.2.1 Message Classes

For **voice messages**, the server **MUST** set the value of the **PidTagMessageClass** property to 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, the **recipient's** phone 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 **MUST** have type IPM.Note.Microsoft.Voicemail.

For **fax messages**, the server **MUST** set the value of the **PidTagMessageClass** property to the following:

- IPM.Note.Microsoft.FAX.CA

For **missed call notifications**, the server MUST set the value of the **PidTagMessageClass** property to the following:

- IPM.Note.Microsoft.Missed.Voice

2.1.2.2 Attachments

For **messages** with audio content, the server MUST add the audio content as a file **attachment** on the message, in accordance with the procedures for attachment handling, as specified in [MS-OXCMSG].

The attachment file MUST be in either the **WAV** file format, the ASF file format, or the MP3 format<1> (see [WAVE] and [MSDN-ASF]).

If in the WAV format, the audio **codec** MUST be either **G.711** a-law, G.711 m-law, **GSM 6.10** (see [G711] and [GSM610]), or linear PCM. If in the ASF file format, the codec MUST be the **WMA 9 Voice** or WMA 2 codec (see [MSDN-WMSSDK]).

In addition to the common **properties** on the attachment, the server MUST set two properties on the attachment, as follows:

- **PidTagAttachLongFilename**: Set to a unique name in the attachment collection of the message. The file name MUST be unique for the attachment order logic specified in section 2.1.2.3 to function properly. 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**: Set to reflect the audio content type of the message.
 - For WMA 9 Voice encoded messages, this MUST be "audio/wma".
 - For GSM 6.10 encoded messages, this MUST be "audio/gsm".
 - For G.711 encoded messages, this MUST be "audio/wav".
 - For MP3 encoded messages, this MUST be "audio/mp3".

In some situations, a client or server can add more than one audio attachment to a given 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 follow the order that specifies the procedure, as specified in section 2.1.2.3.

2.1.2.3 Attachment Order

The server MUST create the **PidTagVoiceMessageAttachmentOrder** property as defined in [MS-OXPROPS] on any **message** that contains audio **attachments**. Both clients and servers SHOULD consult this property to determine the order in which to play any audio attachments, including the trivial case of only one attachment.

The content of the property is a list of **PidTagAttachLongFilename** values for audio file attachments that are to be played as part of the message. The order **MUST** be the reverse 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 semi-colons. Each file name can be prefixed or suffixed with whitespace. The first file name in the list can be preceded by a semi-colon and the last file name in the list can be suffixed with a semi-colon.

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

- vm.wav
- ;vm.wav
- ; vm.wav
- vm.wav
- vm.wav

Or, for example, a message contains three attachments, the **PidTagAttachLongFilename** properties for which 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 **PidTagVoiceMessageAttachmentOrder** are illustrated by, but 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

2.1.2.4 Audio Notes

The client can allow a user to annotate a **voice message** with textual information after it has been delivered to the user's mailbox. For example, some users might find it useful to take note of 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, as specified in [MS-OXPROPS], to the value of that textual information.

2.1.2.5 ASR Data

ASR Data<2> refers to the text transcription of the audio attachment. In an unprotected **voice message**, this is stored in the **PidNameAutomaticSpeechRecognitionData** property. In a protected voice message, it is handled as an attachment instead. Like other attachments in a

rights-managed e-mail message, the attachment is stored in the "Attachment List" storage of the encrypted **BLOB**.

2.1.2.5.1 ASR XML

2.1.2.5.1.1 Simple Types

2.1.2.5.1.1.1 evm:breakWeightType

Represents a coarse classification of the magnitude of a break in the speech data that was processed to obtain the 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>
```

2.1.2.5.1.1.2 evm:confidenceBandType

Represents a coarse classification of a confidence result (that is itself represented as an **evm:zeroToUnityDoubleType**). 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>
```

2.1.2.5.1.1.3 evm:recoErrorType

Represents success, or the types of errors, for voice recognition.

```
<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="other"/>
  </xs:restriction>
</xs:simpleType>
```

2.1.2.5.1.1.4 evm:recoResultType

Represents the result types for voice recognition.

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

2.1.2.5.1.1.5 evm:versionNumberType

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>
```

2.1.2.5.1.1.6 evm:zeroToUnityDoubleType

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>
```

2.1.2.5.1.2 Complex Types

2.1.2.5.1.2.1 evm:recoObjectType

Represents information for a section of the 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>

```

Element	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").
confidence	evm:zeroToUnityDoubleType	Required. Indicates the speech recognition system's confidence in this suggestion.
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 ID of the following element; that is, the next in time order.

Element	Type	Definition
te	xs:time	Required. Indicates the time (measured from the start of the audio) at which the corresponding utterance ends.
ts	xs:time	Required. Indicates the time (measured from the start of the audio) at which the corresponding utterance begins.
wb	xs:breakWeightType	Optional. Indicates the relative weight of the break in the speech.

2.1.2.5.1.3 Elements

2.1.2.5.1.3.1 ASR

The **ASR** element is the root element of the 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 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>

Attribute	Type	Definition
confidence	evm:zeroToUnityDoubleType	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 elements.
confidenceBand	evm:confidenceBandType	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, 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 provide their ID when sending the transcript.<3>

Attribute	Type	Definition
productVersion	evm:versionNumberType	Optional. If present, indicates the version of the software that was used to produce the transcript.<4>
recognitionError	evm:recoErrorType	Optional. If present, provides for a more specific indication of the success or failure of the recognition than does recognitionResult .
recognitionResult	evm:recoResultType	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 always be "1.0.0.0".

2.1.2.5.1.3.2 Break

The **Break** element represents a discontinuity in the semantic content of the recording. For example, the speech might have paused for significantly longer than the typical 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>

```

2.1.2.5.1.3.3 ErrorInformation

The **ErrorInformation** element provides a mechanism for the partner to return more detailed information when the **recognitionError** attribute of the **ASR** element is set to a value other than "success". The content of the element is expected to contain some diagnostic information that can help recipients 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 "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>
```

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.

2.1.2.5.1.3.4 Feature

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.

```
<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:complexType>
</xs:element>
```

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 has to contain a pointer that will allow an application to locate and (with sufficient permission) access the data.

The following table lists supported values of the **Feature class** attribute.

Feature class name	Reference?	Description
Contact	Yes	A personal contact (of the UM-enabled user to whom the voice message was sent). The reference is the Item ID of the contact object, as returned by Exchange Web Services.
Date	Yes	A date. The reference represents a canonical version of the date. This can be in an <code>xs:date</code> format, or a regional format deduced from the recognition language used.
Mailbox	Yes	A mailbox-enabled user. The reference is the primary SMTP address of the user.
PersonName	Yes	A person's name. The reference has the 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 used.

2.1.2.5.1.3.5 Text

The **Text** element represents a portion of the transcript (typically, 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>
```

```
</xs:complexType>
</xs:element>
```

2.2 Protected Voice Message

A protected **voice message**<5> is similar to a **rights-managed e-mail message**, as specified in [MS-OXORMMS]. 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.

2.2.1 Messages

2.2.1.1 Message Classes

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

- **IPM.NOTE.rpmsg.Microsoft.VoiceMail.CA**, for original messages taken with audio content by telephone as a result of call answering.
- **IPM.NOTE.rpmsg.Microsoft.VoiceMail**, for original messages taken with audio content by telephone resulting from all scenarios other than call answering.

2.2.1.2 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, and has the following requirements:

- **PidNameContentClass** MUST be set to "rpmsg.message".
- **PidTagAttachLongFilename** MUST be set to "message.rpmsg".
- **PidTagAttachMimeType** MUST be set to "application/x-microsoft-rpmsg-message".

A protected **voice message** follows this convention. Anon-protected voice message contains one or more audio attachments and voice message preview data in the **PidNameAutomaticSpeechRecognitionData** property [MS-OXPROPS]. 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].

2.2.2 Audio Attachments

Audio attachments carry the audio content of the **voice message**. This section describes how audio attachments are handled with protected voice messages.

2.2.2.1 Encrypted Audio Attachment

When the 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 extension "umrmwav""umrmwma", or "umrmmp3".

The content of **PidTagVoiceMessageAttachmentOrder** 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 attachment file names have the ".umrmwav", ".umrmwma", or ".umrmmp3" extension.

2.2.3 Protected Voice Message Property

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

2.3 UI Configuration

A client application might want to surface an enhanced user interface (UI) for **Message objects** with the **message** classes specified in section 2.1.2.1 for some users and not for others. In addition, the client might want to 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 if it wants 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 2.1.2.1 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 by using the Configuration Information protocol, as specified in [MS-OXOCFG].

The Dictionary Stream object **MUST** be stored in the Inbox **special folder**.

The Dictionary Stream **MUST** have the **PidTagMessageClass** string **property** set on it. The value of the property **MUST** be IPM.Configuration.UMOLK.UserOptions.

The Dictionary **SHOULD** include the following setting. If the setting does not appear in the Dictionary, or the Dictionary does not exist, the following default value **SHOULD** be assumed:

- outlookFlags
 - Name: (string) "outlookFlags"

- Value: (32-bit integer). The least significant bit MUST correspond to whether the client SHOULD surface special UI for message classes specified in section 2.1.2.1, with 1 corresponding to "do surface" and 0 (zero) corresponding to "do not surface." The second-least significant bit MUST correspond to whether the client SHOULD surface telephony configuration UI to the user, with 1 corresponding to "do surface" and 0 (zero) corresponding to "do not surface". Therefore, the value MUST take one of the following four values:
 1. "0x00000000": Neither surface special UI for the message classes described in section 2.1.2.1, nor for telephony configuration purposes.
 2. "0x00000001": Surface special UI for the message classes specified in section 2.1.2.1, but not for telephony configuration purposes.
 3. "0x00000002": Do not surface special UI for the message classes specified in section 2.1.2.1, but do show special UI for telephony configuration purposes.
 4. "0x00000003": Surface special UI for the message classes specified in section 2.1.2.1 and also for telephony configuration purposes.
- Default: (32-bit integer) "0x00000000"

3 Protocol Details

3.1 Client Details

The client role is to display the Unified Messaging objects specified in section 2.1.2.1. 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-OXCMMSG] 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 2.1.2.4) and/or providing a means to automatically play back the audio content of a **message** by using the **attachments** (section 2.1.2.2) and the attachment order information (section 2.1.2.3). For an example of this experience, see section 4.1.2.

3.1.1 Abstract Data Model

See [MS-OXOMSG].

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

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].

4 Protocol Examples

4.1 Playing a Voice Message

Both examples in the following sections 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 and sees that the *outlookFlags* 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 specified in [MS-OXOMSG]. In particular, it allows 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 file, 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 and sees that the *outlookFlags* 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 presses this button, and the client consults the attachment order information on the **message** (section 2.1.2.3) and sees that the value is "vm2.wma;vm1.wma". From this value, it knows that there are two attachments on the Voice Message object with **PidTagAttachLongFilename** properties "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 knows the attachment is encoded with WMA 9 Voice because the **PidTagAttachMimeType** value of the attachment is "audio/wma". After the audio finishes playing, the client downloads "vm1.wma" and plays it in a similar 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 end user decides to use this feature. The client provides an editable area on the screen into which the user can type a set of notes. When the user is finished, the client persists the entered text in the **PidNameAudioNotes property** 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. General security considerations that pertain to the underlying transport apply (see [MS-OXCMSG]).

5.2 Index of Security Parameters

None.

6 Appendix A: Office/Exchange Behavior

The information in this specification is applicable to the following versions of Office/Exchange:

- Microsoft Office 2003
- Microsoft Exchange Server 2003

- Microsoft Office 2007
- Microsoft Exchange Server 2007
- Microsoft Outlook 2010
- Microsoft Exchange Server 2010

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Office/Exchange behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies Office/Exchange does not follow the prescription.

<1> Section 2.1.2.2: Exchange 2003 and Exchange 2007 do not support the MP3 format.

<2> Section 2.1.2.5: ASR Data is not available in Exchange 2003 and Exchange 2007.

<3> Section 2.1.2.5.1.3.1: Exchange 2010 inserts a value of 925712 in transcripts that it generates.

<4> Section 2.1.2.5.1.3.1: Transcripts that are generated by Exchange 2010 Unified Messaging take form "14.nn.nnnn.nnn", with *n* representing digits.

<5> Section 2.1.2.1.1: Protected Voice Mail is not available in Exchange 2003 and Exchange 2007.

Index

- Applicability statement, 7
- Client details, 21
- Examples, 22
- Fields, vendor-extensible, 8
- Glossary, 5
- Index of security parameters, 23
- Informative references, 7
- Introduction, 5
- Messages, 8
 - Protected voice message, 19
 - Transport, 8
 - UI configuration, 20
- Normative references, 6
- Office/Exchange behavior, 23
- Overview, 7
- Preconditions, 7
- Prerequisites, 7
- Protected voice message, 19
- Protocol details, 21
 - Client details, 21
 - Server details, 22
- Protocol examples, 22
- References, 6
 - Informative references, 7
 - Normative references, 6
- Relationship to other protocols, 7
- Security, 23
 - Considerations for implementers, 23
 - Index of security parameters, 23
- Security considerations for implementers, 23
- Server details, 22
- Standards assignments, 8
- Transport, 8
- UI configuration, 20
- Vendor-extensible fields, 8
- Versioning and capability negotiation, 7