Voice Mail and Fax Objects Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

**Support.** For questions and support, please contact dochelp@microsoft.com.
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/4/2008</td>
<td>0.1</td>
<td>Major</td>
<td>Initial Availability</td>
</tr>
<tr>
<td>4/25/2008</td>
<td>0.2</td>
<td>Minor</td>
<td>Revised and updated property names and other technical content.</td>
</tr>
<tr>
<td>6/27/2008</td>
<td>1.0</td>
<td>Major</td>
<td>Initial Release</td>
</tr>
<tr>
<td>8/6/2008</td>
<td>1.01</td>
<td>Minor</td>
<td>Updated references to reflect date of initial release.</td>
</tr>
<tr>
<td>9/3/2008</td>
<td>1.02</td>
<td>Minor</td>
<td>Revised and edited technical content.</td>
</tr>
<tr>
<td>12/3/2008</td>
<td>1.03</td>
<td>Minor</td>
<td>Minor editorial fixes.</td>
</tr>
<tr>
<td>4/10/2009</td>
<td>2.0</td>
<td>Major</td>
<td>Updated technical content and applicable product releases.</td>
</tr>
<tr>
<td>7/15/2009</td>
<td>3.0</td>
<td>Major</td>
<td>Revised and edited for technical content.</td>
</tr>
<tr>
<td>2/10/2010</td>
<td>3.1.1</td>
<td>Minor</td>
<td>Updated the technical content.</td>
</tr>
<tr>
<td>5/5/2010</td>
<td>3.2.0</td>
<td>Minor</td>
<td>Updated the technical content.</td>
</tr>
<tr>
<td>8/4/2010</td>
<td>3.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>11/3/2010</td>
<td>3.3</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>4.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>8/5/2011</td>
<td>4.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/7/2011</td>
<td>4.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>1/20/2012</td>
<td>5.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/27/2012</td>
<td>6.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>6.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>7.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>7.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/26/2013</td>
<td>8.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>8.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>8.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>8.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
</tbody>
</table>

[MS-OXOUM] - v20210817
Voice Mail and Fax Objects Protocol
Copyright © 2021 Microsoft Corporation
Release: August 17, 2021
<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>7/31/2014</td>
<td>8.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>8.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>3/16/2015</td>
<td>9.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>5/26/2015</td>
<td>10.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>9/14/2015</td>
<td>10.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/13/2016</td>
<td>10.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>10.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/24/2018</td>
<td>11.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/1/2018</td>
<td>12.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/22/2021</td>
<td>13.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>8/17/2021</td>
<td>14.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
</tbody>
</table>
# Table of Contents

1 Introduction ......................................................................................... 6  
1.1 Glossary .......................................................................................... 6  
1.2 References ....................................................................................... 7  
1.2.1 Normative References ................................................................... 7  
1.2.2 Informative References .................................................................. 8  
1.3 Overview .......................................................................................... 8  
1.4 Relationship to Other Protocols ....................................................... 8  
1.5 Prerequisites/Preconditions ............................................................. 8  
1.6 Applicability Statement .................................................................... 8  
1.7 Versioning and Capability Negotiation ............................................ 8  
1.8 Vendor-Extensible Fields .................................................................. 9  
1.9 Standards Assignments ..................................................................... 9  

2 Messages ............................................................................................ 10  
2.1 Transport ......................................................................................... 10  
2.2 Message Syntax ................................................................................ 10  
2.2.1 Namespaces ................................................................................. 10  
2.2.2 Voice Message ............................................................................ 10  
2.2.2.1 Message Classes ..................................................................... 10  
2.2.2.2 Attachments ........................................................................... 11  
2.2.2.3 Attachment Order ................................................................... 11  
2.2.2.4 Audio Notes ............................................................................ 11  
2.2.2.5 ASR Data ............................................................................... 12  
2.2.2.5.1 ASR XML Schema Definition ........................................ 12  
2.2.2.5.1.1 Simple Types .................................................................. 14  
2.2.2.5.1.1.1 evm:breakWeightType Simple Type .......................... 14  
2.2.2.5.1.1.2 evm:confidenceBandType Simple Type ..................... 15  
2.2.2.5.1.1.3 evm:recoErrorType Simple Type ................................ 15  
2.2.2.5.1.1.4 evm:recoResultType Simple Type .............................. 16  
2.2.2.5.1.1.5 evm:versionNumberType Simple Type ....................... 17  
2.2.2.5.1.1.6 evm:zeroToUnityDoubleType ................................... 17  
2.2.2.5.1.2 Complex Types .............................................................. 17  
2.2.2.5.1.2.1 evm:recoObjectType Complex Type ......................... 17  
2.2.2.5.1.3 Elements ......................................................................... 18  
2.2.2.5.1.3.1 ASR Element .............................................................. 18  
2.2.2.5.1.3.2 Break Element ............................................................ 19  
2.2.2.5.1.3.3 ErrorInformation ...................................................... 20  
2.2.2.5.1.3.4 Feature Element ........................................................ 20  
2.2.2.5.1.3.5 Text ............................................................................ 21  
2.2.2.5.1.3.6 Information ............................................................... 22  
2.2.3 Protected Voice Message ............................................................. 22  
2.2.3.1 Messages ................................................................................. 22  
2.2.3.1.1 Message Classes .............................................................. 22  
2.2.3.1.2 Message Content .............................................................. 22  
2.2.3.2 Audio Attachments ................................................................. 23  
2.2.3.3 Protected Voice Message Property ...................................... 23  
2.2.4 UI Configuration .......................................................................... 23  
2.2.5 Message Object Properties ......................................................... 24  
2.2.5.1 PidTagSenderPhoneNumber Property .................................. 24  
2.2.5.2 PidNameXSenderPhoneNumber Property ............................... 24  
2.2.5.3 PidTagVoiceMessageDuration Property .............................. 24  
2.2.5.4 PidNameXVoiceMessageDuration Property ........................ 25  
2.2.5.5 PidTagVoiceMessageSenderName Property ......................... 25  
2.2.5.6 PidNameXVoiceMessageSenderName Property ................. 25  
2.2.5.7 PidTagFaxNumberOfPages Property ................................... 25  

[MS-OXOUM] - v20210817  
Voice Mail and Fax Objects Protocol  
Copyright © 2021 Microsoft Corporation  
Release: August 17, 2021
1 Introduction

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

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

- **binary large object (BLOB):** A discrete packet of data that is stored in a database and is treated as a sequence of uninterpreted bytes.
- **codec:** An algorithm that is used to convert media between digital formats, especially between raw media data and a format that is more suitable for a specific purpose. Encoding converts the raw data to a digital format. Decoding reverses the process.
- **Contact object:** A Message object that contains properties pertaining to a contact.
- **dictionary:** A collection of key/value pairs. Each pair consists of a unique key and an associated value. Values in the dictionary are retrieved by providing a key for which the dictionary returns the associated value.
- **fax message:** A fax that a fax server has completely received or transmitted, and archived to the Fax Archive Folder described in [MS-FAX] section 3.1.1.
- **header:** A name-value pair that supplies structured data in an Internet email message or MIME entity.
- **mailbox:** A message store that contains email, calendar items, and other Message objects for a single recipient.
- **message class:** A property that loosely defines the type of a message, contact, or other Personal Information Manager (PIM) object in a mailbox.
- **Message object:** A set of properties that represents an email message, appointment, contact, or other type of personal-information-management object. In addition to its own properties, a Message object contains recipient properties that represent the addressees to which it is addressed, and an attachments table that represents any files and other Message objects that are attached to it.
- **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.
- **Multipurpose Internet Mail Extensions (MIME):** A set of extensions that redefines and expands support for various types of content in email messages, as described in [RFC2045], [RFC2046], and [RFC2047].
- **recipient:** An entity that can receive email messages.
- **rights-managed email message:** An email message that specifies permissions that are designed to protect its content from inappropriate access, use, and distribution.
- **Simple Mail Transfer Protocol (SMTP):** A member of the TCP/IP suite of protocols that is used to transport Internet messages, as described in [RFC5321].
special folder: One of a default set of Folder objects that can be used by an implementation to store and retrieve user data objects.

stream: An element of a compound file, as described in [MS-CFB]. A stream contains a sequence of bytes that can be read from or written to by an application, and they can exist only in storages.

Unified Messaging: A set of components and services that enable voice, fax, and email messages to be stored in a user's mailbox and accessed from a variety of devices.

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

voice message: A Message object that contains audio content recorded by a calling party.

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

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

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

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

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

http://download.microsoft.com/download/7/9/0/790fecaa-f64a-4a5e-a430-0bcccdb3f1b4/ASF_Specification.doc

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?wiki_id=v9jLO9Nb7wSVbbYKNyW

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

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


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

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Reference</th>
</tr>
</thead>
</table>

2.2.2 Voice Message

Voice messages, fax messages, and missed call notifications 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 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.

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 2.2.2.29): 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
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

<!--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:simpleType>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:schema>
```
<xs:restriction base="xs:boolean"/>
</xs:simpleType>
</xs:attribute>
</xs:extension>
</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"/>
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.

```xml
<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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td>A low break weight was used.</td>
</tr>
<tr>
<td>medium</td>
<td>A medium break weight was used.</td>
</tr>
<tr>
<td>high</td>
<td>A high break weight was used.</td>
</tr>
</tbody>
</table>

### 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.

```xml
<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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td>The transcription is of low (possibly poor) quality.</td>
</tr>
<tr>
<td>medium</td>
<td>The transcription is of average quality.</td>
</tr>
<tr>
<td>high</td>
<td>The transcription is of high quality.</td>
</tr>
</tbody>
</table>

### 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.

```xml
<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="messageoolong"/>
    <xs:enumeration value="protectedvoicemail"/>
    <xs:enumeration value="throttled"/>
    <xs:enumeration value="other"/>
    <xs:enumeration value="errorreadingsettings"/>
  </xs:restriction>
</xs:simpleType>
```
The enumerated values for the `recoErrorType` simple type are defined as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>success</td>
<td>The transcription was successfully completed.</td>
</tr>
<tr>
<td>audioQualityPoor</td>
<td>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.</td>
</tr>
<tr>
<td>languageNotSupported</td>
<td>The transcription service cannot process the spoken language used in the voice message.</td>
</tr>
<tr>
<td>rejected</td>
<td>The voice message audio does not conform to the requirements of the transcription system.</td>
</tr>
<tr>
<td>badRequest</td>
<td>The voice message request to the transcription service was not well formed.</td>
</tr>
<tr>
<td>systemError</td>
<td>An unexpected error prevented transcription.</td>
</tr>
<tr>
<td>timeout</td>
<td>The voice transcription process took too long and was stopped.</td>
</tr>
<tr>
<td>messageTooLong</td>
<td>The voice message was too lengthy to be transcribed.</td>
</tr>
<tr>
<td>protectedVoicemail</td>
<td>The voice message has rights protection enabled, and cannot be transcribed.</td>
</tr>
<tr>
<td>throttled</td>
<td>Bandwidth or network limitations prevent this voice message from being transcribed.</td>
</tr>
<tr>
<td>errorReadingSettings</td>
<td>The transcription service cannot read the transcription settings of the user's mailbox.</td>
</tr>
<tr>
<td>other</td>
<td>An unknown error occurred during voice transcription.</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>skipped</td>
<td>The transcription service did not attempt to translate the voice message.</td>
</tr>
<tr>
<td>attempted</td>
<td>The transcription service tried to translate the voice message.</td>
</tr>
</tbody>
</table>
### 2.2.2.5.1.1.5 evm:versionNumberType Simple Type

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

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

### 2.2.2.5.1.1.6 evm:zeroToUnityDoubleType

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

```xml
<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.

```xml
<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: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].

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>be</td>
<td>xs:Boolean</td>
<td>Optional. Indicates whether the element is calculated to be on the most probable (1-best) path through the transcript (if &quot;1&quot; or &quot;true&quot;), or not (if &quot;0&quot; or &quot;false&quot;).</td>
</tr>
<tr>
<td>c</td>
<td>evm:zeroToUnityDoubleType (section 2.2.5.1.1.6)</td>
<td>Required. Indicates the speech recognition system's confidence in this suggestion.</td>
</tr>
<tr>
<td>id</td>
<td>xs:ID</td>
<td>Required. Uniquely identifies the element within the transcript.</td>
</tr>
<tr>
<td>nx</td>
<td>xs:token</td>
<td>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.</td>
</tr>
<tr>
<td>te</td>
<td>xs:time</td>
<td>Required. Indicates the time (measured from the start of the audio) at which the corresponding message ends.</td>
</tr>
<tr>
<td>ts</td>
<td>xs:time</td>
<td>Required. Indicates the time (measured from the start of the audio) at which the corresponding message begins.</td>
</tr>
</tbody>
</table>

### 2.2.5.1.3 Elements

#### 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).

```xml
<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: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:complexType>
</xs:element>
```
The ASR element has the following attributes. Any data types not specified in this document are specified in [XMLSCHEMA2/2].

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>confidence</td>
<td>evm:zeroToUnityDoubleType</td>
<td>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.</td>
</tr>
<tr>
<td>confidenceBand</td>
<td>evm:confidenceBandType</td>
<td>Optional. Provides a general indication of the system's overall confidence in the recognition results.</td>
</tr>
<tr>
<td>lang</td>
<td>xs:language</td>
<td>Required. Indicates the language in which the attempt at automatic speech recognition was made.</td>
</tr>
<tr>
<td>productID</td>
<td>xs:unsignedInt</td>
<td>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&lt;3&gt; provide their ID when sending the transcript.</td>
</tr>
<tr>
<td>productVersion</td>
<td>evm:versionNumberType</td>
<td>Optional. If present, indicates the version of the software that was used to produce the transcript.&lt;4&gt;</td>
</tr>
<tr>
<td>recognitionError</td>
<td>evm:recoErrorType</td>
<td>Required. If present, provides for a more specific indication of the success or failure of the recognition than does the recognitionResult attribute.</td>
</tr>
<tr>
<td>recognitionResult</td>
<td>evm:recoResultType</td>
<td>Required. Indicates whether an attempt at recognition was made and, if so, whether the recognition was completed.</td>
</tr>
<tr>
<td>schemaVersion</td>
<td>evm:versionNumberType</td>
<td>Required. Indicates the version of the schema description. This SHOULD be &quot;1.0.0.0&quot;.</td>
</tr>
</tbody>
</table>

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.

```xml
<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.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>wt</td>
<td>evm:breakWeightType</td>
<td>Optional. Indicates the magnitude of the break.</td>
</tr>
</tbody>
</table>

### 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 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".

```xml
<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].

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>lang</td>
<td>xs:language</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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].

```xml
<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:complexType>
</xs:element>
```
The **Feature** element has the following attributes. Any data types not specified in this document are specified in [XMLSCHEMA2/2].

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>xs:token</td>
<td>Required. Indicates the type of feature that has been identified.</td>
</tr>
<tr>
<td>reference</td>
<td>xs:token</td>
<td>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.</td>
</tr>
<tr>
<td>reference2</td>
<td>xs:token</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Feature class name</th>
<th>Reference?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>Date</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>Mailbox</td>
<td>Yes</td>
<td>A mailbox-enabled user. The reference is the primary Simple Mail Transfer Protocol (SMTP) address of the user.</td>
</tr>
<tr>
<td>PersonName</td>
<td>Yes</td>
<td>A person's name. The reference has the same value as the contained text.</td>
</tr>
<tr>
<td>PhoneNumber</td>
<td>No</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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.

```xml
<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.

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

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>lang</td>
<td>xs:language</td>
<td>Required. Indicates the language used for transcription.</td>
</tr>
<tr>
<td>linkURL</td>
<td>xs:anyURI</td>
<td>Optional. The <strong>URL</strong> where the transcript file can be obtained.</td>
</tr>
<tr>
<td>linkText</td>
<td>xs:normalizedString</td>
<td>Optional. The text for the <strong>linkURL</strong> attribute.</td>
</tr>
</tbody>
</table>

2.2.3 Protected Voice Message

A protected voice message 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 2.2.2.29): 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 by using the Configuration Information Protocol, as specified in [MS-OXOCFG].

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

The dictionary stream 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 SHOULD include the following outlookFlags parameter, which is used to indicate whether the client displays telephony configuration UI and special UI information for message classes. If the outlookFlags parameter does not appear in the dictionary stream or the dictionary stream 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. The four possible values are listed in the following table; the value 0x00000000 is the default.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Display neither the special UI information for message classes nor the telephony configuration UI.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Display only the special UI information for message classes.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Display only the telephony configuration UI.</td>
</tr>
<tr>
<td>0x00000003</td>
<td>Display both the special UI information for message classes and the telephony configuration UI.</td>
</tr>
</tbody>
</table>

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 2.1007) contains the telephone number of the caller associated with a voice message.

The relationship between this property and the X-CallingTelephoneNumber MIME header 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 2.486) contains the telephone number of the caller associated with a voice message.

The relationship between this property and the X-CallingTelephoneNumber MIME header 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 2.1060) specifies the length of the attached voice message, in seconds.

The relationship between this property and the X-VoiceMessageDuration MIME header 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 2.503) specifies the length of the attached voice message, in seconds.

The relationship between this property and the X-VoiceMessageDuration MIME header 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 2.1061) 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 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 2.504) 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 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 2.695) 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 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 2.484) 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 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 2.1059) 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-OXCMMSG] 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`

The relationship between this property and the **X-AttachmentOrder MIME header** 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 2.502) 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** 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 2.628) is a unique identifier associated with the phone call.
The relationship between this property and the MIME header 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 2.483) is a unique identifier associated with the phone call.

The relationship between this property and the MIME header 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 2.375) 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 2.485) 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 2.373) 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 2.2.1. 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-OXCMSP].

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-OXCMSP] section 3.1.4.12. The server MUST set the **PidTagAttachLongFilename** property ([MS-OXCMSP] section 2.2.2.10) and the **PidTagAttachMimeTag** property ([MS-OXCMSP] section 2.2.2.29) 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 2.2.2.29) 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 2.2.15) 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 updates to those products.

- Microsoft Exchange Server 2003
- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Exchange Server 2019
- Microsoft Office Outlook 2003
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016
- Microsoft Outlook 2019
- Microsoft Outlook 2021

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

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

<1> Section 2.2.2.2: 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, Exchange 2013, Exchange 2016, and Exchange 2019 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, Exchange 2013, Exchange 2016, and Exchange 2019 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

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Revision class</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Appendix A: Product Behavior</td>
<td>Updated list of supported products.</td>
<td>major</td>
</tr>
</tbody>
</table>
Playing a voice message example
down-level experience 31
overview 31
up-level experience 31

Preconditions 8
Prerequisites 8
Product behavior 33

Protected voice message
audio attachments 23
protected voice message property 23
Protected Voice Message message 22
Protected voice message property 23

R

References 7
informative 8
normative 7
Relationship to other protocols 8

S

Security
implementer considerations 32
parameter index 32

Sequencing rules
cient 28
server 29

Server
abstract data model 29
initialization 29
message processing 29
other local events 30
overview 29
sequencing rules 29
timer events 30
timers 29

Server - higher-layer triggered events
creating a voice message 29

Standards assignments 9
Syntax 10

T

Timer events
client 28
server 30

Timers
client 28
server 29

Tracking changes 34

Transport 10

Triggered events - client
playing an audio message with multiple
attachments 28

Triggered events - server
creating a voice message 29

U

UI Configuration message 23

V

Vendor-extensible fields 9

Versioning 8

Voice Message message 10

Voice messages
ASR data 12
attachment order 11
attachments 11
audio notes 11
message classes 10