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

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	August 6,	1.01	Updated references to reflect date of initial release.

Corporation	2008		
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.

Table of Contents

1 Introduction

The Voice Mail and Fax Objects protocol provides the methods that are used by the server to manipulate Unified Messaging objects.

1.1 Glossary

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

attachment
folder
message
Message object
property
recipient
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.

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-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.

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

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 or the ASF file format (see [WAVE] and [MSDN-ASF]).

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

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

2. **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 and MUST be ".wma" for files in the ASF format.
   - **PidTagAttachMimeTag**: Set to reflect the audio content type of the message.
     - For WMA 9 Voice encoded messages, this MUST be an "audio/wma".
     - For GSM 6.10 encoded messages, this MUST be "audio/gsm".
     - For G.711 encoded messages, this MUST be "audio/wav".

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.

2.2 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-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 2.1.2.4) 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 **PidTagAttachMimeTag** 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

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.

Index