[MS-OXBBODY]: Best Body Retrieval Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights**. This protocol 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 protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.

- **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 protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp/default.mspx). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting protocol@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.

Preliminary Documentation. This documentation is preliminary documentation for these protocols. Since the documentation may change between this preliminary version and the final version, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Tools. This protocol documentation is intended for use in conjunction with publicly available standard specifications and networking programming art, and assumes that the reader is either familiar with the aforementioned material or has immediate access to it. A protocol specification does not require the use of Microsoft programming tools or programming environments in order for a Licensee to develop an implementation. Licensees who have access to Microsoft programming tools and environments are free to take advantage of them.

Revision Summary			
Author	Date	Version	Comments
Microsoft Corporation	April 4, 2008	0.1	Initial Availability.
Microsoft Corporation	April 25, 2008	0.2	Revised and edited technical content.

Table of Contents

1 Introduction

Servers support three different formats for message text:

- Plain text
- Rich Text Format (RTF)
- HTML

In order to support clients that handle only one body format, servers MUST make any of the three body formats available on demand through alternate body properties. This document specifies how to determine which of the three body formats is the primary or "best" body format.

Exactly how message text is converted from one format to another, and to what extent formatting is preserved in the conversion, is implementation-dependent.

1.1 Glossary

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

HTML
message body
Message object
plain text
Rich Text Format (RTF)

The following data types are defined in [MS-OXCDATA]:

PtypString
PtypBinary
PtypBoolean

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

[MS-OXCDATA] Microsoft Corporation, "Data Structures Protocol Specification", April 2008.

[MS-OXCROPS] Microsoft Corporation, "Remote Operations (ROP) List and Encoding Protocol Specification", April 2008.

[MS-OXGLOS] Microsoft Corporation, "Office Exchange Protocols Master Glossary", April 2008.

[MS-OXOMSG] Microsoft Corporation, "E-mail Object Protocol Specification", April 2008.

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

[MS-OXOSMIME] Microsoft Corporation, "S/MIME E-mail Object Protocol Specification", April 2008.

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

[MS-OXRTFEX] Microsoft Corporation, "Rich Text Format (RTF) Extensions Specification", April 2008.

[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.

1.2.2 Informative References

None.

1.3 Structure Overview (Synopsis)

A client requests a specific **message body** format from a server by issuing a RopGetPropertiesSpecific or RopOpenStream request for the corresponding property. The following table lists the properties that correspond to each body format, plus one important related property.

Body format	Property identifier	Property type	Description
Plain	PidTagBody	PtypString	Message body text in plain text format.
RTF	PidTagRtfCompressed	PtypBinary	Message body text in RTF format.
HTML	PidTagHtml	PtypBinary	Message body text in HTML format.

Body format	Property identifier	Property type	Description
Varies	PidTagRtfInSync	PtypBoolean	Indicates whether PidTagBody and PidTagRtfCompressed contain the same text (ignoring formatting).

Because a server can provide the message body in any of these three formats, an algorithm to determine which is the "original," "primary," or "best" body is needed for clients that are capable of handling multiple body formats. Section 2 of this document specifies that algorithm.

1.4 Relationship to Protocols and Other Structures

This specification relies on [MS-OXCROPS], [MS-OXPROPS], and [MS-OXCDATA] for specification of RopGetPropsSpecific request, property values, and status codes.

1.5 Applicability Statement

The algorithm in section 2.1 applies to **Message objects** of all types except the following:

- The value of PidTagMessageClass is exactly "IPM.Note.SMIME (see section 2.3). (**Note:** If the value of PidTagMessageClass is "IPM.Note.SMIME.MultipartSigned", the algorithm in section 2.1 **is applicable**.)
- The value of PidTagMessageClass is "IPM.Note" and the value of PidNameContentClass is "rpmsg.message (see section 2.4).

1.6 Versioning and Localization

None.

1.7 Vendor-Extensible Fields

None.

2 Structures

2.1 Best Body Algorithm

Step 1. Issue a RopGetPropertiesSpecific request for four properties: **PidTagBody**, **PidTagRtfCompressed**, **PidTagHtml**, **PidTagRtfInSync**. If the status code returned by RopGetPropertiesSpecific indicates a failure, the body type is undefined and the algorithm exits.

If the status code returned by **RopGetPropertiesSpecific** indicates success or warning, then the server MUST have returned four property values in the same order as the requested property tags.

Step 2. Create four variables: PlainStatus, RtfStatus, HtmlStatus, RtfInSync. Examine the returned property values and assign values to the corresponding variables as follows. In each case, if there is an error code it MUST be either NotFound or NotEnoughMemory.

- **PlainStatus** - If the PidTagBody value is String, then assign NoError to PlainStatus; else copy the error code to PlainStatus.
- **RtfStatus** - If the PidTagRtfCompressed value is Binary, then assign NoError to RtfStatus; else copy the error code to RtfStatus.
- **HtmlStatus** - If the PidTagHtml value is Binary, then assign NoError to HtmlStatus; else copy the error code to HtmlStatus.
- **RtfInSync** - If the PidTagRtfInSync value is Boolean, then copy the Boolean value to RtfInSync; else assign FALSE to RtfInSync.

Step 3. Determine the body format based on values of the four variables created in step 2. The following table below can be implemented as an if-then-else chain, in exactly the order specified.

	PlainStatus	**RtfStatus**	**HtmlStatus**	**RtfInSync**	**Body Format**
1	NotFound	NotFound	NotFound	Any	Undefined
2	NotEnoughMemory	NotFound	NotFound	Any	Plain text
3	NotEnoughMemory	NotEnoughMemory	NotFound	Any	RTF
4	NotEnoughMemory	NotEnoughMemory	NotEnoughMemory	True	RTF
5	NotEnoughMemory	NotEnoughMemory	NotEnoughMemory	False	HTML
6	Any	NoError or NotEnoughMemory	NoError or NotEnoughMemory	True	RTF
7	Any	NoError or NotEnoughMemory	NoError or NotEnoughMemory	False	HTML
8	NoError or NotEnoughMemory	NoError or NotEnoughMemory	Any	True	RTF
9	NoError or NotEnoughMemory	NoError or NotEnoughMemory	Any	False	Plain

This table can be implemented in the following pseudocode. Each row of the table is one clause of an ‘if-elseif’ chain. Within a row, each column is ANDed together to form the condition of an ‘if’ clause:

	Code to implement
1	if ((PlainStatus = NotFound) and (RtfStatus = NotFound) and (HtmlStatus = NotFound)) then BodyFormat = Undefined
2	else if ((PlainStatus = NotEnoughMemory) and (RtfStatus = NotFound) and (HtmlStatus = NotFound)) then BodyFormat = Plain
3	else if ((PlainStatus = NotEnoughMemory) and (RtfStatus = NotEnoughMemory) and (HtmlStatus = NotFound)) then BodyFormat = Rtf
4	else if ((PlainStatus = NotEnoughMemory) and (RtfStatus = NotEnoughMemory) and (HtmlStatus = NotEnoughMemory) and (RtfInSync = True)) then BodyFormat = Rtf
5	else if ((PlainStatus = NotEnoughMemory) and (RtfStatus = NotEnoughMemory) and (HtmlStatus = NotEnoughMemory) and (RtfInSync = False)) then BodyFormat = Html
6	else if ((RtfStatus = NoError or RtfStatus = NotEnoughMemory) and (HtmlStatus = NoError or HtmlStatus = NotEnoughMemory) and (RtfInSync = True)) then BodyFormat = Rtf
7	else if ((RtfStatus = NoError or RtfStatus = NotEnoughMemory) and (HtmlStatus = NoError or HtmlStatus = NotEnoughMemory) and (RtfInSync = False)) then BodyFormat = Html
8	else if ((PlainStatus = NoError or PlainStatus = NotEnoughMemory) and (RtfStatus = NoError or RtfStatus = NotEnoughMemory) and (RtfInSync = True)) then BodyFormat = Rtf
9	else if ((PlainStatus = NoError or PlainStatus = NotEnoughMemory) and (RtfStatus = NoError or RtfStatus = NotEnoughMemory) and (RtfInSync = False)) then BodyFormat = Plain

2.2 Plain Text or HTML Converted to RTF

When the result of the algorithm in section 2.1 is RTF, it is possible to determine whether the RTF was generated from original plain text or HTML, as specified in [MS-OXRTFEX].

2.3 Special Considerations for S/MIME Secure Messages

The algorithm of section 2.1 yields an accurate result for a clear-signed S/MIME message (the value of PidTagMessageClass is "IPM.Note.SMIME.MultipartSigned"). Its result is undefined for other types of S/MIME messages (the value of PidTagMessageClass is "IPM.Note.SMIME"). Refer to [MS-OXOSMIME] for a detailed specification of these message types.

2.4 Special Considerations for Rights-Managed Secure Messages

Rights-managed secure messages are messages where the value of PidTagMessageClass is "IPM.Note" and the value of PidNameContentClass is "rpmsg.message", as specified in [MS-OXORMMS].

For rights-managed secure messages, the message body properties specified in this document do not contain the actual message body; instead, they contain boilerplate text intended for recipients whose clients do not support rights-managed secure messages. The actual message body resides in an attachment and is not accessible as a property of the message object. To obtain the actual message body, a client MUST decrypt and parse the attachment as specified in [MS-OXORMMS].

While the algorithm in section 2.1 does yield a result for rights-managed secure messages, that result applies to the boilerplate text and not to the actual message body.

3 Structure Examples

Suppose the following, very simple HTML message is sent to a server:

```
From: <user1@example.com>
To: <user2@example.com>
Subject: test HTML message
Date: Tue, 24 Jan 2006 01:58:57 -0800
MIME-Version: 1.0
Content-Type: text/html
Content-Transfer-Encoding: 7bit
Content-Class: urn:content-classes:message
Importance: normal

<HTML><BODY>Test message, <b>please</b> delete.</BODY></HTML>
```

Then the four property values of interest appear as follows:

Property	Value
PidTagBody	error, NotEnoughMemory
PidTagHtml	<HTML><HEAD><meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"></HEAD><BODY>Test message, please delete.</BODY></HTML>
PidTagRtfCompressed	error, NotEnoughMemory
PidTagRtfInSync	FALSE

The algorithm of section 2.1 creates the four variables shown in the following table:

Property	Value
PlainStatus	NotEnoughMemory
RtfStatus	NotEnoughMemory
HtmlStatus	NoError
RtfInSync	FALSE

Then the algorithm in section 2.1 uses these four newly created variables and matches clause 7:

7	`else if ((RtfStatus = NoError or RtfStatus = NotEnoughMemory) and (HtmlStatus = NoError or HtmlStatus = NotEnoughMemory) and (RtfInSync = False)) then BodyFormat = Html`

And the result returned is HTML body format.

4 Security Considerations

Refer to sections 2.3 and 2.4.

5 Appendix A: Office/Exchange Behavior

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

- Office 2003 with Service Pack 3 applied
- Exchange 2003 with Service Pack 2 applied
- Office 2007 with Service Pack 1 applied

- Exchange 2007 with Service Pack 1 applied

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