[MS-OXOSMIME]: S/MIME E-mail Object 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 Corporation	August 6, 2008	1.01	Revised and edited technical content.

Table of Contents

1 Introduction

This document specifies the details of a **message**'s internal format and describes the mapping between internal format and Internet e-mail format for two specific classes of Internet e-mail messages: messages signed or encrypted according to **S/MIME** standard, and arbitrary **clear-signed message**s that use the "multipart/signed" MIME format.

When the server receives an Internet e-mail message, it maps the message to an internal format known as the **Message object schema**. Similarly, when the client submits an e-mail message via the server, the server maps the message from its internal format to Internet format for sending. Also, in cases where protocols supported by the server allow saving or reading e-mail messages in Internet format, similar mapping is required to and/or from internal format. For more information about the mapping between internal format and Internet format, see [MS-OXCMAIL].

1.1 Glossary

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

> **Attachment object**
> **body part**
> **header field**
> **message**
> **message body**
> **message class**
> **Message object**
> **Message object schema**
> **MIME**
> **MIME entity**
> **MIME message**
> **named property**
> **property**
> **S/MIME**

The following terms are specific to this document:

clear-signed message: An Internet e-mail **message** in the format defined by [RFC1847] and identified with the **media type** "multipart/signed", or the **Message object** representing such a **message**. One important class of **clear-signed message**, based on a "multipart/signed" format, is **S/MIME clear-signed message**, as specified in [RFC3851] and [RFC3852].

Content-Disposition header field: A **MIME header field** specified by [RFC2045].

Content-Transfer-Encoding header field: A **MIME header field** specified by [RFC2045].

Content-Type header field: A **MIME header field** specified by [RFC2045].

encrypted S/MIME message: An Internet e-mail **message** in the format specified by [RFC3851] that uses the EnvelopedData CMS content type [RFC3852], or the **Message object** representing such a **message**.

header field parameter: A name-value pair that provides additional structured information for a **header field**, as specified by [RFC2045].

header of the message: The collection of **header fields** within a **message**, as specified in [RFC2822].

media type: A value in a **Content-Type Header field**, as specified by [RFC2045].

message signature: The signature specified by [RFC3851].

MIME entity body part: A **body part** specified by [RFC2045].

MIME entity header: A type of header specified by [RFC2045].

opaque-signed S/MIME message: An Internet e-mail **message** in the format specified by [RFC3851] that uses the SignedData CMS content type [RFC3852], or the **Message object** representing such a **message**.

RFC2822 message: A **message** in the format specified by [RFC2822].

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-OXCMAIL] Microsoft Corporation, "RFC2822 and MIME to E-mail Object Conversion Protocol Specification", June 2008.

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

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

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

[RFC1847] Galvin, J., Murphy, S., Crocker, S., and Freed, N., "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted", RFC 1847, October 1995, http://www.ietf.org/rfc/rfc1847.txt.

[RFC2045] Freed, N., et al., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996, http://www.ietf.org/rfc/rfc2045.txt.

[RFC2046] Freed, N. and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996,http://www.ietf.org/rfc/rfc2046.txt.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996,http://www.ietf.org/rfc/rfc2047.txt.

[RFC2048] Freed, N., Klensin, J., and Postel, J., "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", RFC 2048, November 1996, http://www.ietf.org/rfc/rfc2048.txt.

[RFC2049] Freed, N. and Borenstein N., "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, November 1996, http://www.ietf.org/rfc/rfc2049.txt.

[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.

[RFC2822] Resnick, P., Ed., "Internet Message Format", RFC 2822, April 2001, http://www.ietf.org/rfc/rfc2822.txt.

[RFC3851] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004, http://www.ietf.org/rfc/rfc3851.txt.

1.2.2   Informative References

[RFC3852] Housley, R. "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004, http://www.ietf.org/rfc/rfc3852.txt.

1.3   Protocol Overview

The client and the server process and represent e-mail **messages** as **Message objects** structured according to the **Message object schema**. For an overview of the **Message** object schema, the **Message** object, the **Attachment object**, and other details of internal format, see [MS-OXCMSG].

A conversion between the Internet e-mail format and the **Message** object schema might be necessary when an incoming message arrives or, in the reverse, when an outgoing message has to be sent in Internet format as specified by [RFC2822], [RFC2045], [RFC2046], [RFC2047], [RFC2048], [RFC2049], [RFC1847], or [RFC3851]. Such a conversion maps **MIME entities** to **Attachment** objects or the **message body**, and maps message **header fields** and **MIME entity header** fields to **properties** of the **Message** object or **Attachment** object. For more details about the entire conversion process, see [MS-OXCMAIL] section 1.3.

This document specifies the special case of such conversion for two specific classes of Internet e-mail messages: arbitrary **clear-signed messages**, and **S/MIME** opaque-signed and encrypted messages. This document only specifies the special handling necessary for these two classes of messages; for more information about the general conversion process (for example, steps that are not unique to just clear-signed messages and/or opaque-signed and encrypted messages), see [MS-OXCMAIL].

Ordinarily, when an **RFC2822 message** or a **MIME message** is mapped to a **Message** object, it is completely deconstructed into a form suitable for direct consumption via a wire protocol, and mappable to a typical client's message presentation. This manner of message deconstruction is not feasible for S/MIME messages for following reasons:

1. Encrypted message content and even the entire message structure are not accessible without a proper decryption key, which is typically not available at delivery time.

2. Signed message content has to be preserved in its entirety, in exactly the form in which was signed, in order for the **message signature** to be verifiable at a later date.

These two points impose restrictions on how the server and the client map an S/MIME message to a **Message** object; general mapping [MS-OXCMAIL] cannot be used without modifications.

A set of mapping conventions exists to resolve this problem and to enable the handling of S/MIME messages as **Message** objects. According to these conventions:

- Unprotected top-level message header fields and MIME entity header fields are mapped to properties of a **Message** object or **Attachment** object in accordance with the general mapping specified in [MS-OXCMAIL].

- The **Message** object is identified as an S/MIME message by having its **message class** property (**PidTagMessageClass**) set to one of the reserved values specified in section 2.2.1 and section 2.2.2.

- The entire protected content of the S/MIME message is mapped to a single **Attachment** object of a corresponding **Message** object.

The following entities can participate in this protocol:

1. Any server or client that wants to represent S/MIME messages through a **Message** object schema.

2. Any client that wants to send or receive S/MIME messages using a server that implements a **Message** object schema.

The S/MIME E-mail Object protocol is limited to top-level **clear-signed** or S/MIME wrapping only; a message classified as clear-signed, opaque-signed, or encrypted can contain other (nested) S/MIME wrapping layers.

This protocol does not distinguish **opaque-signed S/MIME messages** from **encrypted S/MIME messages**.

This document specifies the interpretation and rendering of clear-signed or S/MIME opaque-signed and encrypted messages based on the assumption that the client or server that wants to interpret or render such messages can parse and interpret the corresponding Internet format defined elsewhere [RFC2822], [RFC2045], [RFC2046], [RFC2047], [RFC2048], [RFC2049], [RFC1847], [RFC3851], [RFC3852].

1.4   Relationship to Other Protocols

This protocol defines a special case of mapping between e-mail **messages** in Internet formats [RFC2822], [RFC2045], [RFC2046], [RFC2047], [RFC2048], [RFC2049], [RFC1847], [RFC3851] and a **Message object**. General mapping is specified in [MS-OXCMAIL].

1.5   Prerequisites/Preconditions

None.

1.6   Applicability Statement

This protocol can be utilized by any server or client that wants to represent **S/MIME messages** using a **Message object schema**. It can also be utilized by any client that wants to send or receive S/MIME messages using a server that implements a **Message** object schema.

1.7   Versioning and Capability Negotiation

None.

1.8   Vendor-Extensible Fields

None.

1.9   Standards Assignments

None.

2   Messages

2.1   Transport

None.

2.2   Message Syntax

None.

2.2.1   Clear-Signed Message

A **clear-signed message** in Internet format is a **message** in which the message's **MIME entity** has the **media type** "multipart/signed" as specified in [RFC1847]. Such a MIME entity has two **body parts**: the first part represents signed message content; the second part contains a **message signature**. For more details about multi-part/signed, see [RFC1847].

A clear-signed message in Internet format is mapped to a **Message object** with the following structure:

1. **Message class** SHOULD<1><2><3> be set as "IPM.Note.SMIME.MultipartSigned".
2. **Message body** SHOULD be set by promoting a primary message body MIME entity to appropriate **properties** of a **Message** object as specified by [MS-OXCMAIL]. The method of identifying and promoting a message body is the following:
    a. Consider the first **body part** of a multipart/signed message MIME entity as a complete Internet message.
    b. Apply the heuristics specified in [MS-OXCMAIL] to identify a nested MIME entity as a message body and promote its content according to [MS-OXCMAIL].
3. **Message** object properties other than message class or message body SHOULD be set as specified in [MS-OXCMAIL] and [MS-OXOMSG].
4. The **Message** object MUST contain exactly one **Attachment object**.
    a. Attachment content, stored in the **PidTagAttachDataBinary** property, MUST be set as the entire outer content of a multipart/signed message MIME entity, including a **Content-Type header field** with the value "multipart/signed" and any original parameters. All other message entity **header fields** SHOULD be excluded. It is especially important to preserve the entire original outer content of the first body part within a multipart/signed MIME entity unmodified, as it is protected by a message signature in its original form, and any modification will invalidate the message signature. Note that all message header fields that are excluded are normally processed to populate **Message** object properties as specified in [MS-OXCMAIL].
    b. Other **Attachment** object properties MUST be set as follows:

     i.   **PidTagAttachMethod** MUST be set to a value of "0x00000001" (file attachment).

    ii.   **PidTagAttachMimeTag** MUST be set to a value of "multipart/signed".

   iii.   **PidTagAttachFilename** SHOULD<4>be set to a value of "SMIME.txt".

   iv.   **PidTagAttachLongFilename** SHOULD be set to a value of "SMIME.txt".

    v.   **PidTagDisplayName** SHOULD be set to a value of "SMIME.txt".

   vi.   Other **Attachment** object properties MAY be set as appropriate.

2.2.2 Opaque-Signed and Encrypted S/MIME Message

An **opaque-signed** or encrypted **S/MIME** message in Internet format is identified as a **MIME message** consisting of exactly one **MIME entity**. The MIME entity usually has the **media type** "application/pkcs7-mime" or "application/x-pkcs7-mime", but can alternatively have the media type "application/octet-stream" if a file name, specified by **Content-Type** or **Content-Disposition header field parameters**, has a file extension ".p7m". The content of the entity body is a Cryptographic Message Syntax (CMS) encapsulation of protected **message** content, together with all necessary cryptographic metadata. For more details about CMS, see [RFC3852]. For the purposes of this protocol, the content is treated as opaque binary data. **Message** types specified in [RFC3851] other than opaque-signed or encrypted messages are not supported.

An opaque-signed or **encrypted S/MIME message** in Internet format is mapped to a **Message object** with the following structure:

1. **Message class** SHOULD<5><3> be set as "IPM.Note.SMIME".
2. **Message body** SHOULD NOT be set. Even for an opaque-signed message, where access to message content is possible without possessing a decryption key, the message body SHOULD NOT be promoted to a **Message** object.
3. **Message** object properties other than message class or message body SHOULD be set as specified in [MS-OXCMAIL].
4. The **Message** object SHOULD have a **named property** (with GUID = PS_INTERNET_HEADERS ({00020386-0000-0000-C000-000000000046}) and a string name "Content-Type") that contains the raw ASCII string value of a message MIME entity's Content-Type MIME **header field**, including any parameters of such header field.
5. The message MUST contain exactly one **Attachment object**.
   a. Attachment content, stored in the **PidTagAttachDataBinary** property, MUST be set as the inner content of a message MIME entity. Any Content- Transfer-Encoding applied to a MIME entity body MUST be removed before storing entity body content in an **Attachment** object.
   b. **Attachment** object properties other than content SHOULD be set according to [MS-OXCMAIL], just as they would be if the MIME entity

was a normal message attachment. In particular, the **PidTagAttachMimeTag** property MUST be set to match the media type of a message MIME entity.

3 Protocol Details

3.1 Common Details

3.1.1 Abstract Data Model
None.

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

3.1.5.1 Clear-Signed Message Details

3.1.5.1.1 Recognizing a Clear-Signed Message in Internet Format
The **media type** of the **message MIME entity** is the value of the last **Content-Type header field**. If the message MIME entity's media type is "multipart/signed", then the message SHOULD be treated as a **clear-signed message**. Additional verification steps MAY be performed. For example, a client or server could choose to verify that the multipart/signed MIME entity contains exactly two MIME **body parts** as specified in [RFC1847].

3.1.5.1.2 Converting a Clear-Signed Message in Internet Format into a Message Object
To convert a **clear-signed message** in Internet format into a **Message object**, the following steps SHOULD be performed:

1. From the **message MIME entity**, promote **message header fields** to **Message** object **properties** as specified in [MS-OXCMAIL].
2. Create an **Attachment object**.
3. Set **Attachment** object properties as specified in section 2.2.1.
4. Remove all header fields except the **Content-Type header field** from the message MIME entity,

5. Save the resulting MIME entity as content of the **Attachment** object created in step 2 (for example, set the value of the **PidTagAttachDataBinary** property on the **Attachment** object).

3.1.5.1.3   Recognizing a Message Object that Represents a Clear-Signed Message

If a **Message object** has the **message class** "IPM.Note.SMIME.MultipartSigned" and contains exactly one **Attachment object**, it SHOULD be treated as a **clear-signed message**. Additional verification steps MAY be performed to verify that the **Attachment** object is marked with the appropriate **media type** (e.g. the **PidTagAttachMimeTag** property has a value of "multipart/signed") and represents a valid "multipart/signed" **MIME entity** as specified in [RFC1847]. If the **message** class is not "IPM.Note.SMIME.MultipartSigned", but ends with the suffix ".SMIME.MultipartSigned", the **Message** object MAY <2><3> be treated as a clear-signed message.

If a **Message** object marked with the message class "IPM.Note.SMIME.MultipartSigned" does not have the correct structure specified in section 2.2.1, the behavior is undefined.

3.1.5.1.4   Reconstructing an Internet Format Message from a Clear-Signed Message Object

To reconstruct an Internet format **message** from a **clear-signed Message object**, the following steps SHOULD be performed:
1. Verify that the **Message** object contains exactly one **Attachment object**.
2. Read the **Attachment** object's **PidTagAttachDataBinary** binary **property** value and treat it as a **MIME entity**.
3. Remove all **header fields** except the last **Content-Type header field** from the MIME entity.
4. Add any **message** header fields resulting from promotion of **Message** object **properties** [MS-OXCMAIL] to the MIME entity,.

The resulting MIME entity is a **clear-signed message** in its Internet format. A client or server MAY use a different approach as long as it leads to an equivalent result.

3.1.5.1.5   Reading and Interpreting a Message Object that Represents a Clear-Signed Message

For details about how to recognize a **Message object** representing a **clear-signed message**, see section 3.1.5.1.3.

To read and interpret a clear-signed message, the Internet format SHOULD be reconstructed from a **Message** object as specified in section 3.1.5.1.4. The resulting clear-signed message in its Internet format SHOULD be rendered or interpreted following the guidelines specified in [RFC1847], and possibly [RFC3851], or any other similar specification. A client MAY use a different process as long as it leads to the same rendering or interpretation.

3.1.5.1.6   Composing a New Message Object that Represents a Clear-Signed Message

To compose a new **Message object** that represents a **clear-signed message**, first compose a clear-signed message in its Internet format [RFC1847], and then convert that **message** to a **Message** object as specified in section 3.1.5.1.2. A client MAY use a different process as long as it leads to the same resulting **Message** object content.

3.1.5.2 Opaque-Signed and Encrypted S/MIME Message Details

3.1.5.2.1 Recognizing an S/MIME Opaque-Signed or Encrypted Message in Internet Format

The **media type** of the **message MIME entity** is the value of the last **Content-Type header field**. If the message MIME entity's media type is "application/pkcs7-mime" or "application/x-pkcs7-mime", then the message SHOULD be treated as an **opaque-signed** or encrypted **S/MIME** message. Also, if the message MIME entity's media type is "application/octet-stream", and a file extension specified by the *name* parameter of a **Content-Type header field** or the *filename* parameter of a **Content-Disposition header field** ends with ".p7m" (case-insensitive), the message SHOULD be treated as an opaque-signed or **encrypted S/MIME message**. Additional verification steps MAY be performed. For example, a client or server could choose to verify that MIME entity body content has valid syntax as specified in [RFC3851].

3.1.5.2.2 Converting an S/MIME Opaque-Signed or Encrypted Message in Internet Format into a Message Object

To convert an **S/MIME opaque-signed** or encrypted **message** in Internet format into a **Message object**, the following steps SHOULD be performed:

1. From the message **MIME entity**, promote message **header fields** to **Message** object **properties** as specified in [MS-OXCMAIL].
2. Save the raw ASCII string value of the last **Content-Type header field**, including any parameters of such header, as a **Message** object **named property** with GUID = PS_INTERNET_HEADERS ({00020386-0000-0000-C000-000000000046}) and the name "Content-Type".
3. Promote the message MIME entity as a new **Attachment** object as specified in [MS-OXCMAIL] for a general conversion case.

3.1.5.2.3 Recognizing a Message Object that Represents an S/MIME Opaque-Signed or Encrypted Message

If a **Message object** has the **message class** "IPM.Note.SMIME" and contains exactly one **Attachment object**, it SHOULD be treated as an **S/MIME opaque-signed** or encrypted **message**. Additional verification steps MAY be performed to verify that the **Attachment** object is marked with the appropriate **media type** (e.g. **PidTagAttachMimeTag** is either "application/pkcs7-mime" or "application/x-pkcs7-mime", or it is "application/octet-stream" and *filename*, as specified by **PidTagAttachFilename**, and has a file extension ".p7m") and represents a valid encrypted or opaque-signed message as specified in [RFC3852]. If the message class is not "IPM.Note.SMIME", but ends with the suffix ".SMIME", then the

Message object MAY <4><3> be treated as an S/MIME opaque-signed or encrypted message.

The message class "IPM.Note.SMIME" can be ambiguous . <6>

If a **Message** object marked with the message class "IPM.Note.SMIME" does not have an appropriate structure or content as specified in section 2.2.2, then the behavior is undefined.

3.1.5.2.4 Reconstructing an Internet Format Message from an Opaque-Signed or Encrypted S/MIME Message Object

To reconstruct an Internet format **message** from an **opaque-signed** or encrypted **Message object**, the following steps SHOULD be performed:

1. Verify that the **Message** object contains exactly one **Attachment object**.
2. Create an empty **MIME entity**.
3. Add any message **header fields** resulting from promotion of the **Message** object **properties** [MS-OXCMAIL] to the MIME entity,.
4. Add the **Content-Type header field** to the MIME entity:
    a. If the **Message** object has a **named property** "Content-Type" with GUID PS_INTERNET_HEADERS ({00020386-0000-0000-C000-000000000046}), construct the **Content-Type header field** using the value of the named property, assuming that the value can contain unparsed MIME parameters.
    b. Otherwise, construct the Content-Type header field using a **media type** string obtained from the value of the **Attachment** object's **PidTagAttachMimeTag** property; add a *name* parameter with a value obtained from the **PidTagAttachFilename** property of the **Attachment** object.
5. Add a **clear-signed message** with a disposition value "attachment" to the MIME entity; add a single *filename* parameter with a value obtained from the **PidTagAttachFilename** property of the **Attachment** object, encoded if necessary [MS-OXCMAIL].
6. Add the **Content-Transfer-Encoding header field** with a value of "base64" to the MIME entity.
7. Read the **Attachment** object's **PidTagAttachDataBinary** binary **property** value and encode the result using base64 encoding. Add the result of the encoding as a body of the MIME entity.

The resulting MIME entity is an opaque-signed or encrypted **S/MIME** message in its Internet format. Aclient or server MAY use a different approach as long as it leads to an equivalent result.

3.1.5.2.5 Reading and Interpreting a Message Object that Represents an Opaque-Signed or Encrypted S/MIME Message

For details about how to recognize a **Message object** representing an **S/MIME opaque-signed** or encrypted **message**, see section 3.1.5.2.3.

To read and interpret an S/MIME message, the Internet Format SHOULD be reconstructed from a **Message** object as specified above in section 3.1.5.2.4. The resulting S/MIME message in its Internet format SHOULD be rendered or interpreted following guidelines specified in [RFC3851]. A client MAY use a different process as long as it leads to the same rendering or interpretation.

3.1.5.2.6 Composing a New Message Object that Represents an Opaque-Signed or Encrypted S/MIME Message

To compose a new **Message object** that represents an **S/MIME message**, first compose an **opaque -signed** or **encrypted S/MIME message** in its Internet format [RFC3851], and then convert that message to a **Message** object as specified in section 3.1.5.2.2. A client MAY use a different process as long as it leads to the same resulting **Message** object structure and content.

3.1.6   Timer Events

None.

3.1.7   Other Local Events

None.

4   Protocol Examples

None.

5   Security

5.1   Security Considerations for Implementers

This protocol does not have any security implications beyond those described in [RFC3851]. Furthermore, this protocol treats **S/MIME** content as opaque binary data and does not deal with any sensitive material or data such as encryption keys. Although it is best for clients or servers that render, interpret, or compose S/MIME data to do so in a secure fashion, this is beyond the scope of this specification.

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:

- 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.

<1> Section 2.2.1: In some circumstances, Exchange 2003 SP2 sets the **message class** as "IPM.Note.SMIME". Other clients or servers MUST NOT do this.

<2> Section 2.2.1: Exchange 2003 SP2, Outlook 2003 SP3, and Outlook 2007 SP1 identify any **message** having a **message class** suffix of ".SMIME.MultipartSigned" as a **clear-signed message**. In general though, clients or servers that produce a **Message object** representing a clear-signed message SHOULD NOT assume that others will identify a message having a message class suffix of ".SMIME.MultipartSigned" as a clear-signed message.

<3> Sections 2.2.1, 2.2.2, 3.1.5.2.3: Exchange 2007 SP1 and Outlook 2007 SP1 recognize Office InfoPath signed or encrypted **messages** and, for such messages, they use a dynamically-determined **message class** that starts with the prefix "IPM.InfoPathForm." and ends with the suffix ".SMIME" or ".SMIME.MultipartSigned" [MS-OXCMAIL]. Exchange 2007 SP1 recognizes such message classes as identifying **opaque-signed**, encrypted, or **clear-signed messages**, despite the fact that, in general, it does not recognize other message classes having suffixes ".SMIME" or ".SMIME.MultipartSigned".

<4> Sections 2.2.1, 3.1.5.2.3: Exchange 2007 SP1, Outlook 2003 SP3, and Outlook 2007 SP1 set **PidTagAttachFilename** to a value of "SMIME.p7m".

<5> Section 2.2.2: Exchange 2003 SP2, Outlook 2003 SP3, and Outlook 2007 SP1 set the **message class** to "IPM.Note.Receipt.SMIME" when they identify an S/MIME **message** containsinga secure receipt, as indicated by the *smime-type* parameter with a value of "signed-receipt" on the **Content-Type header field**. Exchange 2003 SP2, Outlook 2003 SP3, and Outlook 2007 SP1 identify any message that has a message class suffix of ".SMIME" as an **opaque-signed** or encrypted message, but other clients/servers SHOULD NOT assume that others will identify a message that has a message class suffix of ".SMIME" as an opaque-signed or encrypted message.

<6> Section 3.1.5.2.3: In Exchange 2003 SP2 only, if a **Message object** has a **message class** of "IPM.Note.SMIME", then the message MAY represent a mislabeled **clear-signed message** with inner **opaque-signed** or encrypted content. This means that, in Exchange 2003 SP2, the

message class "IPM.Note.SMIME" is ambiguous with respect to defining message format. A client or server wishing to interoperate with Exchange 2003 SP2 SHOULD disambiguate the "IPM.Note.SMIME" **Message object** either by analyzing the content of an attachment (for example, the value of the **Attachment object** property **PidTagAttachDataBin**) or by inspecting the value of the **Attachment** object property **PidTagAttachMimeTag**. If the value represents a valid multipart/signed **MIME entity**, then the client or server SHOULD identify the message as a clear-signed message and interpret it according to section 3.1.5.1.

Index