[MS-OXCSPAM]:
Spam Confidence Level Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability.
04/25/2008	0.2		Revised and updated property names and other technical content.
06/27/2008	1.0		Initial Release.
08/06/2008	1.01		Updated references to reflect date of initial release.
09/03/2008	1.02		Revised and edited technical content.
12/03/2008	1.03		Updated IP notice.
03/04/2009	1.04		Revised and edited technical content.
04/10/2009	2.0		Updated applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	3.1.0	Minor	Updated the technical content.
02/10/2010	3.2.0	Minor	Updated the technical content.
05/05/2010	3.3.0	Minor	Updated the technical content.
08/04/2010	3.4	Minor	Clarified the meaning of the technical content.

Contents

1 Introduction

This protocol enables the sharing of preferences for handling the filtering of unsolicited e-mail messages between the client and the server.

This protocol enables the client to process e-mail messages that are likely to be **phishing messages** or **spam** by doing the following:

- Identifying messages that are potentially spam.
- Identifying messages that are potentially phishing messages.
- Blocking the delivery of messages that are from specific senders or classes of senders.
- Allowing the delivery of messages that are either from specific senders or to specific recipients, regardless of whether the messages are identified as spam or phishing messages.

1.1 Glossary

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

domain
entry ID
extended rule
Folder object
Junk E-mail folder
Junk E-Mail rule
mailbox
Message object
named property
phishing
phishing message
property (1)
rule
Simple Mail Transfer Protocol (SMTP)
spam
spam confidence level
spam filter
special folder

The following terms are specific to this document:

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

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. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

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

[MS-OXCFOLD] Microsoft Corporation, "Folder Object Protocol Specification", April 2008.

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

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

[MS-OXORSS] Microsoft Corporation, "RSS Object Protocol Specification", April 2008.

[MS-OXORULE] Microsoft Corporation, "E-Mail Rules Protocol Specification", April 2008.

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

[MS-OXPHISH] Microsoft Corporation, "Phishing Warning Protocol Specification", April 2008.

[MS-OXPROPS] Microsoft Corporation, "Exchange Server Protocols Master Property List", 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

[RFC4406] Lyon, J., and Wong, M., "Sender ID: Authenticating E-Mail", RFC 4406, April 2006, ftp://ftp.rfc-editor.org/in-notes/rfc4406.txt

[RFC4408] Wong, M., and Schlitt, W., "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1", RFC 4408, April 2006, ftp://ftp.rfc-editor.org/in-notes/rfc4408.txt

1.2.2 Informative References

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

1.3 Overview

This protocol enables the sharing of preferences for handling **spam filtering** functionality between the client and the server.

This protocol enables the client to process e-mail messages that are likely to be phishing messages or spam by doing the following:

- Identifying messages that are potentially spam.
- Identifying messages that are potentially phishing messages.
- Blocking the delivery of messages to the Inbox that are from specific senders or classes of senders.
- Allowing the delivery of messages that are either from specific senders or to specific recipients, regardless of whether the messages are identified as spam or phishing messages.

When an e-mail message is delivered to a server, the server executes the **extended rule** that determines where the message is delivered. At the messaging client's discretion, it uses **properties** on the **Junk E-Mail rule** to control the server action and to store information about client spam and **phishing** preferences.

This protocol does not specify any algorithms for determining whether the message is spam or a phishing message; it only specifies how properties on a message are used to determine where the message is delivered.

1.4 Relationship to Other Protocols

This protocol uses properties on the **Message object** as a way to identify messages that are likely to be spam or a phishing message. In addition, this protocol uses message **rules** for the processing of these properties. This protocol also uses properties on **Folder objects** and **special folders**. Therefore, this protocol specification relies on the following:

- An understanding of Folder objects, as specified in [MS-OXCFOLD].
- An understanding of properties, as specified in [MS-OXPROPS], and of how to get and set properties, as specified in [MS-OXCMSG].
- An understanding of Message objects, as specified in [MS-OXOMSG].
- An understanding of e-mail rules, as specified in [MS-OXORULE].
- An understanding of special folders, as specified in [MS-OXOSFLD].

1.5 Prerequisites/Preconditions

This protocol assumes that a system is in place to set and retrieve the properties that are identified in this specification on the e-mail Message objects, on the e-mail rules, and on Folder objects.

1.6 Applicability Statement

This protocol defines the properties and rules to process spam and phishing messages. This protocol does not specify the algorithm that determines the likelihood of a message being spam or a phishing message or whether to consider a sender safe or blocked.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Message properties are transported between the client and server as specified in [MS-OXCMSG]. Message rules are created as defined in [MS-OXORULE].

2.2 Message Syntax

The properties in the following sections are specific to this protocol. This protocol references commonly used data types, as specified in [MS-OXCDATA].

2.2.1 Message Object Properties

The properties listed in the following sections are persisted on a Message object.

2.2.1.1 PidLidSpamOriginalFolder

Type: **PtypBinary**

If present, this property indicates which folder a message was in before it was filtered into the **Junk E-mail folder**. The value of this property is the **entry ID** of the folder that contained the message before it was moved (PidTagParentEntryId). This property SHOULD<1> be set when a message is marked as spam.

2.2.1.2 PidNameExchangeJunkEmailMoveStamp

Type: **PtypInteger32**, unsigned

If present and valid, this property indicates that the message MUST NOT be processed by the client's spam filter because the message either was already processed or the message is safe. The stamp is valid only if it matches the Junk E-Mail Move Stamp, as specified in section 3.1.4.1. If present and invalid, this property MUST be ignored and the message MUST be processed by the client's spam filter. If this property is not present, then the message MUST be processed by the client's spam filter.

2.2.1.3 PidTagContentFilterSpamConfidenceLevel

Type: **PtypInteger32**, signed

This property SHOULD<2> be stamped by a spam filter before the Junk E-Mail rule is executed. This value indicates a confidence level that the message is spam and MUST be in the range 0xFFFFFFFF to 0x00000009 (inclusive). A value of 0xFFFFFFFF indicates that the message is to be considered "not spam", and a value greater than 0xFFFFFFFF indicates the likelihood that the message is spam; the higher the number, the higher the likelihood that the message is spam.

2.2.1.4 PidTagSenderIdStatus

Type: **PtypInteger32**, unsigned

A server MUST set this property to report the results of a Sender-ID check, as defined in [RFC4406]. This property MUST have the values listed in the following table, which correspond to the definitions in [RFC4408].

Symbolic name	Value
Neutral	0x00000001
Pass	0x00000002
Fail	0x00000003
SoftFail	0x00000004
None	0x00000005
TempError	0x80000006
PermError	0x80000007

2.2.2 Junk E-Mail Rule Properties

The properties listed in the following sections are persisted on the Junk E-Mail rule.

2.2.2.1 PidTagJunkAddRecipientsToSafeSendersList

Type: **PtypInteger32**, unsigned

If present, this property MUST be set to 0x00000000 or 0x00000001. A value of 0x00000001 indicates that the mail recipients are to be added to the safe senders list. A value of 0x00000000 indicates that the mail recipients are not to be added to the safe senders list.

2.2.2.2 PidTagJunkIncludeContacts

Type: **PtypInteger32**, unsigned

This property indicates whether e-mail addresses of the contacts in the Contacts folder are treated in a special way with respect to the spam filter.

If set to 0x00000001, these e-mail addresses MUST populate the "trusted" contact e-mail address portion of the Junk E-Mail rule restriction, as specified in section 3.1.4.2, such that mail from these addresses is treated as "not junk". If set to 0x00000000, e-mail addresses from the Contacts folder MUST NOT be added to the Junk E-Mail rule, and the section of the rule MUST be NULL. See section 3.1.4.2 for more details.

2.2.2.3 PidTagJunkPermanentlyDelete

Type: **PtypInteger32**, unsigned

If set to 0x00000001, messages identified as spam MAY be permanently deleted. If set to 0x00000000, then messages identified as spam MUST NOT be permanently deleted.

2.2.2.4 PidTagJunkPhishingEnableLinks

Type: **PtypBoolean**

If the value is 0x01, then the phishing stamp on the message, as specified in [MS-OXPHISH] section 2.2.1.1, SHOULD be ignored. If the value is 0x00, then the phishing stamp on the message SHOULD NOT be ignored.

2.2.2.5 PidTagJunkThreshold

Type: **PtypInteger32**, unsigned

This property indicates how aggressively the client is to send incoming mail to the Junk E-mail folder. It corresponds to the high/low/none filter setting. A value of 0xFFFFFFFF indicates that spam filtering SHOULD NOT be applied; however, block lists MUST still be applied. A value of 0x80000000 indicates that all mail is spam except those messages from senders on the trusted senders list or sent to recipients on the trusted recipients list.

The following table lists the values for this property.

No spam filtering	0xFFFFFFFF
Low spam filtering	0x00000006
High spam filtering	0x00000003
Trusted Lists Only	0x80000000

2.2.2.6 PidTagReportTime

Type: **PtypTime**

This property indicates the last time the contact list that is controlled by PidTagJunkIncludeContacts was updated.

2.2.3 Inbox Folder Properties

The property listed in the following section is on the Inbox folder.

2.2.3.1 PidTagAdditionalRenEntryIds

Type: **PtypMultipleBinary**

This property is persisted on the Inbox folder of a message store, as specified in [MS-OXOSFLD]. The value at zero-based index five is used to validate that the PidNameExchangeJunkEmailMoveStamp property that is stamped on a message was stamped by this message store. It MUST be read and used as specified in section 3.1.4.1.

3 Protocol Details

3.1 Server Details

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.

3.1.1.1 Junk E-Mail Move Stamp

A valid PidNameExchangeJunkEmailMoveStamp property, when stamped on a message, indicates that the message bypasses the client's spam filter.

A common scenario in which this occurs is when the client's spam filter has already moved the message to the Junk E-mail folder once. If the user has retrieved a message from the Junk E-mail folder, it will not be reprocessed.

If clients want to populate a message store with trusted Message objects that are never spam but might look like spam to a spam filter, they can set this property. The RSS Object Protocol [MS-OXORSS] is a practical example of this method.

3.1.1.2 Junk E-Mail Rule

The Junk E-Mail rule stores preferences regarding how spam filtering is applied.

The format of the preferences is a server-side extended rule that follows the E-Mail Rules Protocol as specified in [MS-OXORULE]. This format is convenient for a server that implements the E-Mail Rules Protocol, as executing the rule on a message will apply the spam filtering preferences to the message and move it to the Junk E-mail folder if it fits the condition for spam.

The rule can be created or maintained by either the client or the server, but the rule itself is executed only on the server. That is, no client-side operations are associated with the Junk E-Mail rule.

The restriction that makes up the condition of the Junk E-Mail rule, as specified in [MS-OXORULE] section 2.2.1.3.2.9, contains several interdependent clauses. These clauses are essentially lists of **Simple Mail Transfer Protocol (SMTP)** e-mail addresses and several categories of e-mail **domains**.

The following table lists these clauses.

Blocked Sender Addresses	E-mail addresses of senders (who the message was sent FROM) to be blocked.
Blocked Sender Domains	E-mail domains "@bad.com" of senders that can be blocked.
Trusted Sender Domains	E-mail domains "@good.com" of senders that are trusted.
Trusted Recipient Domains	E-mail domains "@good.com" of recipients (who the message was sent TO) that are trusted.

Blocked Sender Addresses	E-mail addresses of senders (who the message was sent FROM) to be blocked.
Trusted Sender Addresses	E-mail addresses of senders that can be trusted.
Trusted Recipient Addresses	E-mail addresses of recipients that can be trusted.
Trusted Contact Addresses	E-mail addresses of contacts from the **mailbox** Contacts folder.

There is also a clause that checks the value of the PidTagContentFilterSpamConfidenceLevel property, in the event that this property was applied to the message during delivery.

In the event that the received message "fails" the restriction, the following happens:

1. The message is moved to the Junk E-mail folder.
2. The message is stamped with the PidNameExchangeJunkEmailMoveStamp property.

3.1.2 Timers

None.

3.1.3 Initialization

The **Junk E-Mail Move Stamp** and Junk E-Mail rule SHOULD be created on the first interaction of the user with a mailbox.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Obtaining or Creating the Junk E-Mail Move Stamp

The Junk E-Mail Move Stamp, PidNameExchangeJunkEmailMoveStamp, is stamped on every message that is moved by the Junk E-Mail rule or is otherwise trusted content.

A PidNameExchangeJunkEmailMoveStamp is only valid if it matches the value in PidTagAdditionalRenEntryIds, as specified in section 3.1.4.1.1.

3.1.4.1.1 Obtaining the Junk E-Mail Move Stamp

To obtain the value of the Junk E-Mail Move Stamp, the client MUST read the PidTagAdditionalRenEntryIds property from the Inbox folder and do one the following:

- If there is a value at zero-based index 5 of the array, this value is the value of the PidNameExchangeJunkEmailMoveStamp property, stored as an unsigned **PtypInteger32**. The client MUST use this value for the PidNameExchangeJunkEmailMoveStamp property when creating the Junk E-Mail rule.
- If there is no value at zero-based index 5, the client MUST generate a value for the PidNameExchangeJunkEmailMoveStamp property, as described in section 3.1.4.1.2.

3.1.4.1.2 Generating the Junk E-Mail Move Stamp

If there is no value at zero-based index 5 of the PidTagAdditionalRenEntryIds property of the Inbox folder, the client MUST generate an arbitrary **PtypInteger32** value and store it as an unsigned **PtypInteger32** to the zero-based index 5 of the PidTagAdditionalRenEntryIds property of the Inbox folder. See section 5.1.1 for security details.

The client MUST set the value of the PidNameExchangeJunkEmailMoveStamp property to the **PtypInteger32** value that is stored in the zero-based index 5 of the PidTagAdditionalRenEntryIds property of the Inbox folder.

3.1.4.2 Creating the Junk E-Mail Rule

The Junk E-Mail rule or "spam" rule is a server-side extended rule that follows the E-Mail Rules Protocol, as specified in [MS-OXORULE]. The client MUST create and maintain the rule in the following prescribed format.

The rule MUST be created in the Associated Contents folder of the Inbox folder.

The PidTagRuleMessageName property MUST be set to "Junk E-Mail rule".

The PidTagSubject property MUST be set to "Junk E-Mail rule".

The PidTagRuleMessageProvider property MUST be set to "JunkEmailRule".

The PidTagRuleMessageState property MUST be set to ST_ENABLED | ST_EXIT_LEVEL | ST_SKIP_IF_SCL_IS_SAFE.

The PidTagRuleMessageSequence property MUST be set to 0 (zero).

The PidTagRuleMessageUserFlags property MUST be set to 0 (zero).

The PidTagRuleMessageLevel property MUST be set to 0 (zero).

The PidTagExtendedRuleMessageActions property MUST contain two actions:

- An OP_MOVE action to the Junk E-mail folder.
- An OP_TAG action to stamp the moved message with the **named property**, with the value of the PidNameExchangeJunkEmailMoveStamp.

The restriction elements that are used in this and subsequent sections, such as RES_AND, FL_IGNORECASE, and so on, are specified in [MS-OXCDATA].

E-mail addresses MUST be Simple Mail Transfer Protocol (SMTP) e-mail addresses.

The rule condition restriction that is set on property PidTagExtendedRuleMessageCondition MUST have the following format:

- A RES_AND restriction with two sub-clauses
  1. A RES_OR restriction with two sub-clauses
     - 1. A RES_OR restriction with zero or more sub-clauses, one for each "bad" sender e-mail address. Each restriction MUST be of the format:
       - A RES_CONTENT restriction with a ulFuzzyLevel of FL_SUBSTRING | FL_IGNORECASE ([MS-OXCDATA] section 2.13.4.1) comparing the value of property

PidTagSenderEmailAddress with a string that contains the e-mail address of a "bad" sender; for example, "bad-user@example.com"

- 2. A RES_AND restriction with two sub-clauses
  1. A RES_OR restriction with two sub-clauses
     1. A RES_AND restriction with two sub-clauses

        1. A RES_EXIST restriction for property PidTagContentFilterSpamConfidenceLevel
        2. A RES_PROPERTY for property PidTagContentFilterSpamConfidenceLevel, with a relative operation of RELOP_GT against a value of -1.

     1. A RES_OR restriction with zero or more sub-clauses, one for each "bad" sender domain. Each restriction MUST be of the format:
        - A RES_CONTENT restriction with a ulFuzzyLevel of FL_SUBSTRING | FL_IGNORECASE ([MS-OXCDATA] section 2.13.4.1) comparing the value of property PidTagSenderEmailAddress with a string that contains the domain of a "bad" sender, for example "@bad-domain.com"
     2. A RES_NOT restriction with one sub-clause
        - 1. A RES_OR restriction with two sub-clauses

          1. A RES_OR restriction with zero or more sub-clauses, one for each "trusted" sender domain. Each restriction MUST be of the following format:

             A RES_CONTENT restriction with a ulFuzzyLevel of FL_SUBSTRING | FL_IGNORECASE ([MS-OXCDATA] section 2.13.4.1) that compares the value of property PidTagSenderEmailAddress with a string that contains the domain of a trusted sender; for example, "@good-domain.com"

          2. A RES_SUB restriction for property PidTagMessageRecipients, with the sub-clause

             - A RES_OR restriction with zero or more sub-clauses, one for each "trusted" recipient domain. Each restriction MUST be of the format:
               - A RES_CONTENT restriction with a ulFuzzyLevel of FL_SUBSTRING | FL_IGNORECASE ([MS-OXCDATA] section 2.13.4.1) that compares the value of property PidTagEmailAddress with a string that contains the domain of a trusted recipient; for example, "@good. domain.com"

2. A RES_NOT restriction with one sub-clause
   - 1. A RES_OR restriction with three sub-clauses
     1. A RES_OR restriction with zero or more sub-clauses, one for each "trusted" sender e-mail address. Each restriction MUST be of the format:
        - A RES_CONTENT restriction with a ulFuzzyLevel of FL_SUBSTRING | FL_IGNORECASE ([MS-OXCDATA] section 2.13.4.1) that compares the value of property PidTagSenderEmailAddress with a string that contains the e-mail address of a trusted sender; for example, "good-user@example.com",

2. A RES_SUB restriction for property PidTagMessageRecipients, with the sub-clause

   - A RES_OR restriction with zero or more sub-clauses, one for each "trusted" recipient e-mail address. Each restriction MUST be of the format:

     - A RES_CONTENT restriction with a ulFuzzyLevel of FL_SUBSTRING | FL_IGNORECASE ([MS-OXCDATA] section 2.13.4.1) that compares the value of property PidTagEmailAddress with a string that the e-mail address of a trusted recipient, for example "good-user@example.com"

3. A RES_OR restriction with zero or more sub-clauses. Each restriction MUST be of the format:

   - A RES_CONTENT restriction with a ulFuzzyLevel of FL_SUBSTRING | FL_IGNORECASE ([MS-OXCDATA] section 2.13.4.1) that compares the value of property PidTagSenderEmailAddress with a string that contains the e-mail address of a contact from the mailbox's contact list, for example, "user1@example.com". If property PidTagJunkIncludeContacts is set to 0x00000000, this restriction SHOULD be empty (NULL); if the PidTagJunkIncludeContacts property is set to 0x00000001, then there SHOULD be one of these restrictions one for each "trusted" contact e-mail address.

The properties PidTagReportTime, PidTagJunkIncludeContacts, and PidTagJunkThreshold MUST be set as specified in section 2.

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 Client Details

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.

3.2.1.1 Junk E-Mail Move Stamp

This is as specified in section 3.1.1.1.

3.2.1.2 Junk E-Mail Rule

The Junk E-Mail rule stores preferences regarding how spam filtering occurs for this mailbox.

Clients do not implement the E-Mail Rules Protocol, as specified in [MS-OXORULE], but still use the rule to store user preferences. Clients interpret properties on the message and the data in PidTagExtendedRuleMessageCondition as specifying preferences and lists of data that are used to control the filter and the spam user interfaces elements.

The rule contains a PidTagContentFilterSpamConfidenceLevel, which can be used as the user preference for how aggressively spam is filtered.

The rule contains various lists of SMTP e-mail addresses that are stored in the PidTagExtendedRuleMessageCondition. For a summary of these, see section 3.1.1.2.

The client can use these lists and preferences to control a client-side spam filter and user interface elements, and also to communicate preferences to the server.

3.2.2 Timers

None.

3.2.3 Initialization

The Junk E-Mail Move Stamp and Junk E-Mail rule SHOULD be created on the first interaction of the user with a mailbox that requires them.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Obtaining or Creating the Junk E-Mail Move Stamp

The client MUST obtain or create the Junk E-Mail Move Stamp as specified in section 3.1.4.1.

3.2.4.2 Creating the Junk E-Mail Rule

The client MUST create the Junk E-Mail rule whenever it requires the spam preferences as specified in section 3.1.4.2.

Although the client does not execute rules, the client MUST format the Junk E-mail message as specified in section 3.1.4.2.

3.2.4.3 Retrieval of Spam Preferences

After clients log on to the messaging server, they SHOULD retrieve preferences from the Junk E-Mail rule before they apply any spam filtering on messages.

3.2.4.4 User Changes Client Spam Preferences

When users change their spam preferences, messaging clients SHOULD update the Junk E-Mail rule to match these new preferences.

3.2.4.5 Server Junk E-Mail Rule Changes

Clients SHOULD recognize when the server Junk E-Mail rule changes.

3.2.4.6 User Adds a New Contact to Their Contacts Folder

If PidTagJunkIncludeContacts is present with a value of 0x00000001, and if the added contact has e-mail addresses that are not yet included in the trusted contacts section of the Junk E-Mail rule, those e-mail addresses MUST be added to the restriction.

If PidTagJunkIncludeContacts is 0x00000000, no action is required.

3.2.4.7 User Sends an E-Mail

If PidTagJunkAddRecipientsToSafeSendersList is present with a value of 0x00000001, the SMTP addresses of the e-mail recipients MUST be added to trusted senders clause of the Junk E-Mail rule condition.

If PidTagJunkAddRecipientsToSafeSendersList is 0x00000000, no action is required.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Receiving a Message

3.2.5.1.1 Receiving a Message using Spam Filtering

If the client chooses to run a spam filter to determine if the message is spam, the client SHOULD use the preferences specified in the Junk E-Mail rule to control the spam filter actions.

3.2.5.1.2 Receiving a Message with PidNameExchangeJunkEmailMoveStamp

If the client receives a message that has the PidNameExchangeJunkEmailMoveStamp property (section 2.2.1.2) set, that property MUST be validated against the PidTagAdditionalRenEntryIds property (section 2.2.3.1), as specified in section 3.1.4.1. If the value matches, the client MUST NOT run a spam filter against this message.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Adding a Sender to the Trusted Recipients List

Jim consistently receives mail from a mailing list that his spam filter moves to the Junk E-mail folder. Jim trusts all mail sent to the mailing list, and so adds the mailing list SMTP address "recip2@example.com" to his trusted recipients list.

The client first receives a handle to the Junk E-mail message by using RopOpenMessage.

The client retrieves property PidTagExtendedRuleMessageCondition by using RopGetPropertiesSpecific. The response contains the following data:

```
0000: 00 00 00 02 00 00 00 01-02 00 00 00 01 03 00 00
0010: 00 03 00 00 01 00 1F 00-1F 0C 1F 00 1F 0C 62 00
0020: 6C 00 6F 00 63 00 6B 00-65 00 64 00 32 00 40 00
0030: 65 00 78 00 61 00 6D 00-70 00 6C 00 65 00 2E 00
0040: 63 00 6F 00 6D 00 00 00-03 00 00 01 00 1F 00 1F
0050: 0C 1F 00 1F 0C 62 00 6C-00 6F 00 63 00 6B 00 65
0060: 00 64 00 33 00 40 00 65-00 78 00 61 00 6D 00 70
0070: 00 6C 00 65 00 2E 00 63-00 6F 00 6D 00 00 00 03
0080: 00 00 01 00 1F 00 1F 0C-1F 00 1F 0C 62 00 6C 00
0090: 6F 00 63 00 6B 00 65 00-64 00 40 00 65 00 78 00
00a0: 61 00 6D 00 70 00 6C 00-65 00 2E 00 63 00 6F 00
00b0: 6D 00 00 00 00 02 00 00-00 01 02 00 00 00 00 02
00c0: 00 00 00 08 03 00 76 40-04 02 03 00 76 40 03 00
00d0: 76 40 FF FF FF FF 01 00-00 00 00 02 01 02 00 00
00e0: 00 01 01 00 00 00 03 01-00 01 00 1F 00 1F 0C 1F
00f0: 00 1F 0C 40 00 65 00 78-00 61 00 6D 00 70 00 6C
0100: 00 65 00 2E 00 63 00 6F-00 6D 00 00 00 09 0D 00
0110: 12 0E 01 00 00 00 00 02-01 03 00 00 00 01 01 00
0120: 00 00 03 00 00 01 00 1F-00 1F 0C 1F 00 1F 0C 73
0130: 00 61 00 66 00 65 00 40-00 65 00 78 00 61 00 6D
0140: 00 70 00 6C 00 65 00 2E-00 63 00 6F 00 6D 00 00
0150: 00 09 0D 00 12 0E 01 01-00 00 00 03 00 00 01 00
0160: 1F 00 03 30 1F 00 03 30-72 00 65 00 63 00 69 00
0170: 70 00 40 00 65 00 78 00-61 00 6D 00 70 00 6C 00
0180: 65 00 2E 00 63 00 6F 00-6D 00 00 00 01 00 00 00
0190: 00
```

The following table lists the spam lists that this data corresponds to. In the "C-style string representation" column, the letter "L" that precedes each string literal indicates that the string is a wide-character string literal (that is, an array of wchar_t).

List	C-style string representation
Blocked Sender Addresses	L"blocked@example.com" L"blocked2@example.com" L"blocked3@example.com"
Blocked Sender Domains	None
Trusted Sender Domains	L "@example.com"
Trusted recipient Domains	None

List	C-style string representation
Trusted Sender Addresses	L"safe@example.com"
Trusted Recipient Addresses	L"recip@example.com"
Trusted Contact Addresses	None

The client constructs the new restriction, including recip2@example.com as a trusted recipient. The client sets the new property value on the message. Because this condition can be large, the client chooses to set the property by calling RopOpenStream, RopSetStreamSize, RopWriteStream, RopCommitStream, and RopRelease. The RopWriteStream sets the following data:

```
0000: 00 00 00 02 00 00 00 01-02 00 00 00 01 03 00 00
0010: 00 03 00 00 01 00 1F 00-1F 0C 1F 00 1F 0C 62 00
0020: 6C 00 6F 00 63 00 6B 00-65 00 64 00 32 00 40 00
0030: 65 00 78 00 61 00 6D 00-70 00 6C 00 65 00 2E 00
0040: 63 00 6F 00 6D 00 00 00-03 00 00 01 00 1F 00 1F
0050: 0C 1F 00 1F 0C 62 00 6C-00 6F 00 63 00 6B 00 65
0060: 00 64 00 33 00 40 00 65-00 78 00 61 00 6D 00 70
0070: 00 6C 00 65 00 2E 00 63-00 6F 00 6D 00 00 00 03
0080: 00 00 01 00 1F 00 1F 0C-1F 00 1F 0C 62 00 6C 00
0090: 6F 00 63 00 6B 00 65 00-64 00 40 00 65 00 78 00
00a0: 61 00 6D 00 70 00 6C 00-65 00 2E 00 63 00 6F 00
00b0: 6D 00 00 00 00 02 00 00-00 01 02 00 00 00 00 02
00c0: 00 00 00 08 03 00 76 40-04 02 03 00 76 40 03 00
00d0: 76 40 FF FF FF FF 01 00-00 00 00 02 01 02 00 00
00e0: 00 01 01 00 00 00 03 01-00 01 00 1F 00 1F 0C 1F
00f0: 00 1F 0C 40 00 65 00 78-00 61 00 6D 00 70 00 6C
0100: 00 65 00 2E 00 63 00 6F-00 6D 00 00 00 09 0D 00
0110: 12 0E 01 00 00 00 00 02-01 03 00 00 00 01 01 00
0120: 00 00 03 00 00 01 00 1F-00 1F 0C 1F 00 1F 0C 73
0130: 00 61 00 66 00 65 00 40-00 65 00 78 00 61 00 6D
0140: 00 70 00 6C 00 65 00 2E-00 63 00 6F 00 6D 00 00
0150: 00 09 0D 00 12 0E 01 02-00 00 00 03 00 00 01 00
0160: 1F 00 03 30 1F 00 03 30-72 00 65 00 63 00 69 00
0170: 70 00 32 00 40 00 65 00-78 00 61 00 6D 00 70 00
0180: 6C 00 65 00 2E 00 63 00-6F 00 6D 00 00 00 03 00
0190: 00 01 00 1F 00 03 30 1F-00 03 30 72 00 65 00 63
01a0: 00 69 00 70 00 40 00 65-00 78 00 61 00 6D 00 70
01b0: 00 6C 00 65 00 2E 00 63-00 6F 00 6D 00 00 00 01
01c0: 00 00 00 00
```

This data corresponds to the spam lists in the following table.

List	C-style string representation
Blocked Sender Addresses	L"blocked@example.com" L"blocked2@example.com" L"blocked3@example.com"
Blocked Sender Domains	None
Trusted Sender Domains	L "@example.com"

List	C-style string representation
Trusted Recipient Domains	None
Trusted Sender Addresses	L"safe@example.com"
Trusted Recipient Addresses	L"recip@example.com" L"recip2@example.com"
Trusted Contact Addresses	None

Finally, the client sends a RopSaveChangesMessage request to persist the object on the server, and a RopRelease request to release the object.

5 Security

5.1 Security Considerations for Implementers

5.1.1 Junk E-Mail Move Stamp Security Considerations

As specified in section 2.2.1.2, PidNameExchangeJunkEmailMoveStamp is used to bypass content protection offered by client spam filters. If the valid value of the Junk E-Mail Move Stamp can be determined by an outside party, that party might discover a clever way to exploit the protocol such that untrusted and potentially malicious content could bypass protective filters.

Implement section 3.1.4.1.2 in such a way that the value of the zero-based index 5 of the PidTagAdditionalRenEntryIds property of the Inbox folder cannot be guessed.

5.2 Index of Security Parameters

Security Parameter	Section
PidNameExchangeJunkEmailMoveStamp	2.2.1.2

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products:

- Microsoft® Office Outlook® 2003
- Microsoft® Exchange Server 2003
- Microsoft® Office Outlook® 2007
- Microsoft® Exchange Server 2007
- Microsoft® Outlook® 2010
- Microsoft® Exchange Server 2010

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product 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 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 product does not follow the prescription.

<1> Section 2.2.1.1: Exchange 2003, Exchange 2007, and Exchange 2010 do not set PidLidSpamOriginalFolder.

<2> Section 2.2.1.3: Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not set this property on messages.

7 Change Tracking

This section identifies changes that were made to the [MS-OXCSPAM] protocol document between the May 2010 and August 2010 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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 or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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 **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type "Editorially updated."

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.1 Glossary	57303 Added "Junk E-mail folder" to the list of terms that are defined in [MS-OXGLOS].	N	Content update.
1.2.1 Normative References	55751 Moved [MS-OXGLOS] from Normative References section to Informative References section.	N	Content update.
1.2.1 Normative References	56767 Added a reference for [MS-OXPROPS].	N	Content update.
1.4 Relationship to Other Protocols	56767 Added reference to [MS-OXPROPS].	N	Content update.

8 Index