[MS-OXCSPAM]:
Spam Confidence Level Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

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

- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

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.
11/03/2010	3.5	Minor	Clarified the meaning of the technical content.
03/18/2011	3.5	No change	No changes to the meaning, language, and formatting of the technical content.

Table of 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.

1.1 Glossary

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

domain

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

contact
entry ID
extended rule
Folder object
Inbox folder
Junk E-mail folder
mailbox
Message object
named property
phishing
phishing message
rule
Simple Mail Transfer Protocol (SMTP)
spam
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-OXCMSG] Microsoft Corporation, "Message and Attachment Object Protocol Specification", June 2008.

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

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

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

[MS-OXPHISH] Microsoft Corporation, "Phishing Warning Protocol Specification", June 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

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

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

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

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

[MS-OXORSS] Microsoft Corporation, "RSS Object Protocol Specification", June 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:

- 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 the e-mail **Message object**, on the e-mail **rules**, and on **Folder objects** 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 described in [MS-OXCFOLD].

- An understanding of properties, as described in [MS-OXPROPS], and of how to get and set properties, as described in [MS-OXCMSG].

- An understanding of Message objects, as described in [MS-OXOMSG].

- An understanding of e-mail rules, as described in [MS-OXORULE].

- An understanding of special folders, as described 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 that are relevant to the processing of 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**, as defined in [MS-OXPROPS] section 2.951). 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 be stamped by the server 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.

2.2.2 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. When the value is 0xFFFFFFFF, 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** ([MS-OXPROPS] section 2.855) 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 ([MS-OXPROPS] section 2.457) 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

This is as specified in section 3.2.1.1.

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 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.
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 ([MS-OXPROPS] section 2.718), in the event that this property was applied to the message during delivery.

For more details about executing the Junk E-Mail rule on a message, see section 3.1.5.1.

3.1.2 Timers

None.

3.1.3 Initialization

None.

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**, as specified in section 2.2.1.2) SHOULD be created on the first interaction of the user with a mailbox. The Junk E-Mail Move Stamp is stamped on every message that is moved by the Junk E-Mail rule or is otherwise trusted content. For details about obtaining or creating the Junk E-Mail Move Stamp, see sections 3.2.4.1.1 and 3.2.4.1.2.

3.1.4.2 Creating the Junk E-Mail Rule

The Junk E-Mail rule or "spam" rule SHOULD be created on the first interaction of the user with a mailbox. It is a server-side extended rule that follows the E-Mail Rules Protocol, as specified in [MS-OXORULE], and MUST be created and maintained in the following prescribed format.

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

The **PidTagRuleMessageName** property ([MS-OXORULE] section 2.2.4.1.1) MUST be set to "Junk E-Mail rule".

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

The **PidTagRuleMessageProvider** property ([MS-OXORULE] section 2.2.4.1.7) MUST be set to "JunkEmailRule".

The **PidTagRuleMessageState** property ([MS-OXORULE] section 2.2.4.1.4) MUST be set to ST_ENABLED | ST_EXIT_LEVEL | ST_SKIP_IF_SCL_IS_SAFE.

The **PidTagRuleMessageSequence** property ([MS-OXORULE] section 2.2.4.1.3) MUST be set to 0 (zero).

The **PidTagRuleMessageUserFlags** property ([MS-OXORULE] section 2.2.4.1.5) MUST be set to 0 (zero).

The **PidTagRuleMessageLevel** property ([MS-OXORULE] section 2.2.4.1.6) MUST be set to 0 (zero).

The **PidTagExtendedRuleMessageActions** property ([MS-OXORULE] section 2.2.4.1.9) 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** (section 2.2.1.2).

The restriction elements that are used in this and subsequent sections, such as RES_AND and FL_IGNORECASE, 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** ([MS-OXORULE] section 2.2.4.1.10) MUST have the following format:

A RES_AND restriction with two subclauses:

1. A RES_OR restriction with two subclauses:

   1. A RES_OR restriction with zero or more subclauses, one for each "bad" sender e-mail address. Each restriction MUST be a RES_CONTENT restriction with a ulFuzzyLevel of FL_FULLSTRING | FL_IGNORECASE ([MS-OXCDATA] section 2.12.4.1) comparing the value of property **PidTagSenderEmailAddress** ([MS-OXOMSG] section 2.2.1.41) 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 subclauses:

      1. A RES_OR restriction with two subclauses:

         ▪ A RES_AND restriction with two subclauses:

            1. A RES_EXIST restriction for property **PidTagContentFilterSpamConfidenceLevel** (section 2.2.1.3).

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

         ▪ A RES_OR restriction with zero or more subclauses, one for each "bad" sender domain. Each restriction MUST be a RES_CONTENT restriction with a ulFuzzyLevel of FL_SUBSTRING | FL_IGNORECASE ([MS-OXCDATA] section 2.12.4.1) 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 a subclause RES_OR restriction with two subclauses:

         1. A RES_OR restriction with zero or more subclauses, one for each "trusted" sender domain. Each restriction MUST be a RES_CONTENT restriction with a ulFuzzyLevel of FL_SUBSTRING | FL_IGNORECASE 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 a RES_OR restriction with zero or more sub-clauses, one for each "trusted" recipient domain. Each restriction MUST be a RES_CONTENT restriction with a ulFuzzyLevel of FL_SUBSTRING | FL_IGNORECASE 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 a RES_OR restriction with three subclauses:

   1. A RES_OR restriction with zero or more subclauses, one for each "trusted" sender e-mail address. Each restriction MUST be a RES_CONTENT restriction with a ulFuzzyLevel of FL_FULLSTRING | FL_IGNORECASE 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 a RES_OR restriction with zero or more subclauses, one for each "trusted" recipient e-mail address. Each restriction MUST be a RES_CONTENT restriction with a ulFuzzyLevel of FL_FULLSTRING | FL_IGNORECASE 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 subclauses. Each restriction MUST be a RES_CONTENT restriction with a ulFuzzyLevel of FL_SUBSTRING | FL_IGNORECASE 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** (section 2.2.2.2) is set to 0x00000000, this restriction MUST 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** (section 2.2.2.6), **PidTagJunkIncludeContacts**, and **PidTagJunkThreshold** (section 2.2.2.5) MUST be set as specified in section 2.

3.1.5   Message Processing Events and Sequencing Rules

None.

3.1.5.1   Executing the Junk E-Mail Rule on a Message

When the server executes the Junk E-Mail rule on a message, it applies the spam filtering preferences to the message and then handles the message according to the value of the **PidTagExtendedRuleMessageCondition** property on the Junk E-Mail rule (as specified in section 3.1.4.2).

If the **PidTagExtendedRuleMessageCondition** property on the Junk E-Mail rule evaluates to true, then the server does the following:

1. Moves the message to the Junk E-mail folder.

2. Stamps the message with the **PidNameExchangeJunkEmailMoveStamp** property (section 2.2.1.2).

If the **PidTagExtendedRuleMessageCondition** property on the Junk E-Mail rule evaluates to false, the server routes the message to the Inbox folder.

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

A valid **PidNameExchangeJunkEmailMoveStamp** property (section 2.2.1.2), 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, as described in [MS-OXORSS], is a practical example of this method.

3.2.1.2 Junk E-Mail Rule

The Junk E-Mail rule is a server-side extended rule, as specified in section 3.1.4.2. The client uses the rule to store user preferences regarding how spam filtering occurs for this mailbox, but the rule itself is executed only on the server.

Clients interpret properties on the message and data in the **PidTagExtendedRuleMessageCondition** property ([MS-OXPROPS] section 2.769) of the Junk E-Mail rule as specifying preferences and lists of data that are used to control the client spam filter and user interface elements.

The rule contains a **PidTagContentFilterSpamConfidenceLevel**, ([MS-OXPROPS] section 2.718) 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.

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 Junk E-Mail Move Stamp, **PidNameExchangeJunkEmailMoveStamp** (section 2.2.1.2), is stamped on every message that is moved by the Junk E-Mail rule or is otherwise trusted content.

A **PidNameExchangeJunkEmailMoveStamp** is valid only if it matches the value in **PidTagAdditionalRenEntryIds**, as specified in section 3.2.4.1.1.

The client MUST obtain or create the Junk E-Mail Move Stamp as specified in sections 3.2.4.1.1 and 3.2.4.1.2.

3.2.4.1.1   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 ([MS-OXPROPS] section 2.586) 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 ([MS-OXPROPS] section 2.457), 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 3.2.4.1.2.

3.2.4.1.2   Generating the Junk E-Mail Move Stamp

If there is no value at zero-based index 5 of the **PidTagAdditionalRenEntryIds** property ([MS-OXPROPS] section 2.586) 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** ([MS-OXPROPS] section 2.457) to the **PtypInteger32** value that is stored in the zero-based index 5 of the **PidTagAdditionalRenEntryIds** property of the Inbox folder.

3.2.4.2   Creating the Junk E-Mail Rule

The client MUST create the Junk E-Mail rule, as specified in [MS-OXORULE] section 3.1.4.3.2, whenever it requires the spam preferences as specified in section 3.1.4.2.

The client MUST format the Junk E-Mail rule 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 server Junk E-Mail rule, as specified in [MS-OXORULE] section 3.1.4.2, 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 server Junk E-Mail rule (as specified in section 3.1.4.2 and in [MS-OXORULE] section 3.1.4.3.2) to match these new preferences.

3.2.4.5   Server Junk E-Mail Rule Changes

Clients SHOULD periodically retrieve the server Junk E-Mail rule. as specified in [MS-OXORULE] section 3.1.4.2, and compare the Junk E-Mail rule settings with the client spam filter settings to determine whether the server Junk E-Mail rule has changed. If the Junk E-Mail rule settings do not match the client spam filter settings, the client SHOULD update its **spam filter** settings to match the server Junk E-Mail rule settings. The interval at which the client retrieves the server Junk E-Mail rule and compares the settings is implementation-dependent.

3.2.4.6 User Adds a New Contact to Their Contacts Folder

If **PidTagJunkIncludeContacts** ([MS-OXPROPS] section 2.855) is present with a value of 0x00000001, the client MUST determine whether the added contact has e-mail addresses that are not yet included in the trusted contacts section of the Junk E-Mail rule. If the added contact's e-mail addresses are already included in the trusted contacts section of the Junk E-Mail rule, no action is required. If the added contact has e-mail addresses that are not yet included in the trusted contacts section of the Junk E-Mail rule, the client MUST update the server Junk E-Mail rule (as specified in section 3.1.4.2 and in [MS-OXORULE] section 3.1.4.3.2) to add those e-mail addresses to the restriction.

If **PidTagJunkIncludeContacts** is 0x00000000, no action is required.

3.2.4.7 User Sends an E-Mail

If **PidTagJunkAddRecipientsToSafeSendersList** ([MS-OXPROPS] section 2.854) is present with a value of 0x00000001, the client MUST update the server Junk E-Mail rule (as specified in section 3.1.4.2 and in [MS-OXORULE] section 3.1.4.3.2) to add the SMTP addresses of the e-mail recipients to the 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.

If the client spam filter determines that the message is spam, the client uses the value of the **PidTagJunkPermanentlyDelete** property on the Junk E-Mail rule (as specified in section 2.2.2.3) to determine whether to permanently delete the message.

If the client spam filter determines that the message is spam and moves the message to the Junk E-mail folder, then the client SHOULD set the **PidLidSpamOriginalFolder** property as specified in section 2.2.1.1.

The client can use the **PidTagJunkPhishingEnableLinks** property on the Junk E-Mail rule (as specified in section 2.2.2.4) to determine whether to enable links within the message.

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 rule by using **RopOpenMessage** ([MS-OXCROPS] section 2.2.6.1).

The client retrieves property **PidTagExtendedRuleMessageCondition** ([MS-OXPROPS] section 2.769) by using **RopGetPropertiesSpecific** ([MS-OXCROPS] section 2.2.8.3). 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 00-02-01 03 00 00 00 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"

List	C-style string representation
Trusted recipient Domains	None
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 the following:

- **RopOpenStream** ([MS-OXCROPS] section 2.2.9.1)

- **RopSetStreamSize** ([MS-OXCROPS] section 2.2.9.6)

- **RopWriteStream** ([MS-OXCROPS] section 2.2.9.3)

- **RopCommitStream** ([MS-OXCROPS] section 2.2.9.4)

- **RopRelease** ([MS-OXCROPS] section 2.2.15.3)

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"
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** ([MS-OXCROPS] section 2.2.6.3) 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 3.2.4.1.2 in such a way that the value of the zero-based index 5 of the **PidTagAdditionalRenEntryIds** property ([MS-OXPROPS] section 2.586) 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 or supplemental software. References to product versions include released service packs:

- Microsoft® Exchange Server 2003

- Microsoft® Exchange Server 2007

- Microsoft® Exchange Server 2010

- Microsoft® Office Outlook® 2003

- Microsoft® Office Outlook® 2007

- Microsoft® Outlook® 2010

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. 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 that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2.2.1.1: Exchange 2003, Exchange 2007, and Exchange 2010 do not set **PidLidSpamOriginalFolder** ([MS-OXPROPS] section 2.300).

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index