[MS-OXCSPAM]: Spam Confidence Level, Allow and Block Lists Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

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.

Table of Contents

1 Introduction

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

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

- Identifying messages that are potentially spam

- Identifying messages that are potentially phishing messages

- Blocking delivery of messages that are from specific senders or classes of senders

- Allowing 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]:

binary large object (BLOB)
domain
Folder object
Message object
phishing
phishing message
property
Simple Mail Transfer Protocol (SMTP)
spam
spam confidence level (SCL)
spam filter

The following term is specific to this document:

Junk E-Mail Rule: A server side extended rule that follows the E-Mail Rules protocol documented in [MS-OXORULE] and the properties of which specify preferences for a **spam filter**.

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-DTYP] Microsoft Corporation, "Windows Data Types", March 2007, http://go.microsoft.com/fwlink/?LinkId=111558.

[MS-OXCDATA] Microsoft Corporation, "Data Structures Protocol Specification", 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-OXGLOS] Microsoft Corporation, "Office Exchange Protocols Master Glossary", April 2008.

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt.

[RFC4406] Lyon, J. and Wong, M. "Sender ID: Authenticating E-Mail", RFC 4406, April 2006, http://www.ietf.org/rfc/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, http://www.ietf.org/rfc/rfc4408.txt.

1.2.2 Informative References

None.

1.3 Protocol Overview (Synopsis)

This protocol enables sharing 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** or **spam** in the following ways:

- Identifying messages that are potentially spam

- Identifying messages that are potentially **phishing messages**

- Blocking delivery of messages to the Inbox that are from specific senders or classes of senders

- Allowing 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 which 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 phish 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 a message is delivered.

1.4 Relationship to Other Protocols

This protocol uses properties on the Message object as a means of identifying 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 the Folder object, as specified in [MS-OXCFOLD]

- An understanding of getting and setting properties, as specified in [MS-OXCMSG].

- An understanding of the Message object, 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 presumes that a system is in place to set and retrieve the properties identified in this protocol on the e-mail messages, in the e-mail rules, and on folders.

1.6 Applicability Statement

This protocol defines the properties and rules that MUST be set 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 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.

2.2.1 Message Object Properties

The properties 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 being 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 a spam filter because the message was either 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.

2.2.1.3 PidTagContentFilterSpamConfidenceLeve

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. The higher the number, the higher the likelyhood that the e-mail is spam. A value of -1 indicates that the message is to be considered "not 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 following values, which correspond the definitions in [RFC4408]:

Symbolic name	Value
Neutral	1
Pass	2
Fail	3
SoftFail	4
None	5
TempError	0x80000006
PermError	0x80000007

2.2.2 Junk E-Mail Rule Properties

These properties 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 specially 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 described 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<3> be permanently deleted.

2.2.2.4 PidTagJunkPhishingEnableLinks

Type: PtypBoolean

If TRUE, the phishing stamp on the message, as specified in [MS-OXPHISH],SHOULD be ignored.

2.2.2.5 PidTagJunkThreshold

Type: PtypInteger32, unsigned

This property indicates how aggressively incoming mail SHOULD be sent 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.

Values for this are as follows:

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 controlled by PidTagJunkIncludeContacts was updated.

2.2.3 Inbox Folder Properties

The properties in the following sections are 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 [MX-OXOSFLD]. The value at zero based index five is used to validate that the PidNameExchangeJunkEmailMoveStamp stamped on a message was stamped by this message store. It MUST be read and used as described 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 The Junk E-Mail Move Stamp

A valid Junk E-Mail Move Stamp property, when stamped on a message, indicates that a message bypasses the spam filter.

Typically this occurs because the 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. Protocol [MS-OXORSS] is a practical example of this method.

3.1.1.2 The 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 documented in [MS-OXORULE]. This format is convenient for a server implementing 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, there are no client side operations associated with the Junk E-Mail Rule.

The restriction that makes up the condition of the Junk E-Mail Rule [MS-OXORULE] contains several interdependent clauses. These clauses are essentially lists of SMTP e-mail address and e-mail **domains** of several categories. These include:

Blocked Sender Addresses	E-mail addresses of senders (who the message was sent FROM) that should 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 PidTagContentFilterSpamConfidenceLeve property, in the event 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 Junk E-Mail Move Stamp 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 described 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 do the following:

1. Read the PidTagAdditionalRenEntryIds property from the Inbox folder.
2. If there is a value at zero based index 5 of the array, this value is the value of the Junk E-Mail Move stamp, stored as an unsigned PtypInteger32. The client MUST use this value for the Junk E-Mail Move Stamp property when creating the Junk E-Mail Rule.
3. If there is no value at zero based index 5, the client MUST generate a value for the Junk E-Mail Move Stamp 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, the client MUST generate an arbitrary PtypInteger32 value for the Junk E-Mail Move Stamp. See section 5.1.1 for security details.

The new value of the Junk E-Mail Move stamp MUST be stored as an unsigned PtypInteger32 to 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 which 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 PidTagRuleMsgName property MUST be set to "Junk E-Mail Rule".
The PidTagSubject MUST be set to "Junk E-Mail Rule"
The PidTagRuleMsgProvider 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 PidTagRuleMsgSequence property MUST be set to 0;
The PidTagRuleMsgUserFlags property MUST be set to 0;
The PidTagRuleMsgLevel property MUST be set to 0;
The PidTagRuleMsgActions property MUST contain two actions:

1. An OP_MOVE action to the Junk E-mail folder.
2. An OP_TAG action to stamp the moved message with the named property, with the value of the Junk E-Mail Move Stamp.

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

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

The rule condition restriction set on property PidTagRuleMsgCondition 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_FULLSTRING | FL_IGNORECASE comparing the value of property PidTagSenderEmailAddress with a string containing the e-mail address of a "bad" sender, 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 PidTagContentFilterSpamConfidenceLeve
                    (2) A RES_PROPERTY for property PidTagContentFilterSpamConfidenceLeve, with a relative operation of RELOP_GT against a value of -1.
                (2) 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_FULLSTRING | FL_IGNORECASE comparing the value of property PidTagSenderEmailAddress with a string containing the domain of a "bad" sender, 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 format:

A RES_CONTENT restriction with a ulFuzzyLevel of FL_FULLSTRING | FL_IGNORECASE comparing the value of property PidTagSenderEmailAddress with a string containing the domain of a trusted sender, 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_FULLSTRING | FL_IGNORECASE comparing the value of property PidTagEmailAddress with a string containing the domain of a trusted recipient, 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_FULLSTRING | FL_IGNORECASE comparing the value of property PidTagSenderEmailAddress with a string containing the e-mail address of a trusted sender, 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_FULLSTRING | FL_IGNORECASE comparing the value of property PidTagEmailAddress with a string containing the e-mail address of a trusted recipient, example "good-user@example.com"

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

A RES_CONTENT restriction with a ulFuzzyLevel of FL_FULLSTRING | FL_IGNORECASE comparing the value of property PidTagSenderEmailAddress with a string containing the e-mail address of a contact from the mailbox's contact list, example user1@example.com. If property PidTagJunkIncludeContacts is set to 0x00000000, this restriction SHOULD be empty (NULL).

The PidTagReportTime, PidTagJunkIncludeContacts, and PidTagJunkThreshold properties MUST be set as described 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 The Junk E-Mail Move Stamp

This is as described in section 3.1.1.1.

3.2.1.2 The 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 uses the rule to store user preferences. Clients interpret properties on the message and the data in PidTagRuleMsgCondition as specifying preferences and lists of data used to control the filter and the spam user interfaces elements.

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

The rule contains various lists of SMTP e-mail addresses stored in the PidTagRuleMsgCondition. A summary of these lists is described in 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 with 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.

Though the client doesn't execute rules, the client MUST format the Junk E-mail Message as specified in section3.1.4.2.

3.2.4.3 Retrieval of Spam Preferences

After logging onto the messaging server, clients SHOULD retrieve preferences from the Junk E-Mail Rule before applying any spam filtering on messages.

3.2.4.4 User Changes Client Spam Preferences

When a user changes 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 set, that property MUST be validated against the store stamp value according to 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 using RopOpenMessage.

The client retrieves property PidTagRuleMsgCondition 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
```

This data corresponds to the following spam lists:

List	C-style string representation
Blocked Sender Addresses	L"blocked@example.com"
	L"blocked2@example.com"

19 of 23

[MS-OXCSPAM] - v0.2
Spam Confidence Level, Allow and Block Lists Protocol Specification
Copyright © 2008 Microsoft Corporation.
Release: Friday, April 25, 2008

	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"
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 following lists:

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 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 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.

Implementers of the protocol SHOULD implement section 3.1.4.1.2 in such a way that the value of the Junk E-Mail Move Stamp cannot be guessed.

5.2 Index of Security Parameters

Security Parameter	Section
PidNameExchangeJunkEmailMoveStamp	2.2.1.2

6   Appendix A: Office/Exchange Behavior

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

- Microsoft Office 2003 with Service Pack 3 applied
- Microsoft Exchange 2003 with Service Pack 2 applied
- Microsoft Office 2007 with Service Pack 1 applied
- Microsoft 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> Exchange (all versions) does not set PidLidSpamOriginalFolder
<2> Outlook (all versions) does not set this property on messages.
<3> Exchange (all versions) does not permanently delete mail based on this property.

Index