[MS-OXCSPAM]:

Spam Confidence Level Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1		Initial Availability.
4/25/2008	0.2		Revised and updated property names and other technical content.
6/27/2008	1.0		Initial Release.
8/6/2008	1.01		Updated references to reflect date of initial release.
9/3/2008	1.02		Revised and edited technical content.
12/3/2008	1.03		Updated IP notice.
3/4/2009	1.04		Revised and edited technical content.
4/10/2009	2.0		Updated applicable product releases.
7/15/2009	3.0	Major	Revised and edited for technical content.
11/4/2009	3.1.0	Minor	Updated the technical content.
2/10/2010	3.2.0	Minor	Updated the technical content.
5/5/2010	3.3.0	Minor	Updated the technical content.
8/4/2010	3.4	Minor	Clarified the meaning of the technical content.
11/3/2010	3.5	Minor	Clarified the meaning of the technical content.
3/18/2011	3.5	No change	No changes to the meaning, language, and formatting of the technical content.
8/5/2011	4.0	Major	Significantly changed the technical content.
10/7/2011	4.0	No Change	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	5.0	Major	Significantly changed the technical content.
4/27/2012	5.1	Minor	Clarified the meaning of the technical content.
7/16/2012	5.1	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	5.2	Minor	Clarified the meaning of the technical content.
2/11/2013	5.2	No Change	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	5.3	Minor	Clarified the meaning of the technical content.
11/18/2013	5.3	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	5.3	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	5.3	No Change	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	5.3	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	5.3	No Change	No changes to the meaning, language, or formatting of the technical content.
3/16/2015	6.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Spam Confidence Level Protocol enables the sharing of preferences for the filtering of unsolicited e-mail messages between the client and the server.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

action: A discrete operation that is executed on an incoming **Message object** when all conditions in the same **rule** are TRUE. A rule contains one or more actions.

contact: A person, company, or other entity that is stored in a directory and is associated with one or more unique identifiers and attributes (2), such as an Internet message address or login name.

Contacts folder: A **Folder object** that contains Contact objects.

domain: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a domain controller (DC) and host a member list that identifies all members of the domain, as well as optionally hosting the Active Directory service. The domain controller provides authentication (2) of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

entry ID: See EntryID.

extended rule: A **rule** that is added to, modified, and deleted from a server by using a mechanism other than standard rules, but is otherwise functionally identical to a standard rule.

folder associated information (FAI): A collection of **Message objects** that are stored in a Folder object and are typically hidden from view by email applications. An FAI Message object is used to store a variety of settings and auxiliary data, including forms, views, calendar options, favorites, and category lists.

Folder object: A messaging construct that is typically used to organize data into a hierarchy of objects containing Message objects and **folder associated information (FAI)** Message objects.

Inbox folder: A special folder that is the default location for **Message objects** received by a user or resource.

Junk Email folder: A special folder that is the default location for **Message objects** that are determined to be junk email by a Junk Email rule.

Junk Email rule: An **extended rule** that describes a **spam filter**.

mailbox: A **message store** that contains email, calendar items, and other **Message objects** for a single recipient.

Message object: A set of properties that represents an email message, appointment, contact, or other type of personal-information-management object. In addition to its own properties, a Message object contains recipient properties that represent the addressees to which it is addressed, and an attachments table that represents any files and other Message objects that are attached to it.

message store: A unit of containment for a single hierarchy of Folder objects, such as a mailbox or public folders.

phishing: The luring of sensitive information, such as passwords or other personal information, from a recipient by masquerading as someone who is trustworthy and has a real need for such information.

phishing message: An email message that is designed to trick a recipient into divulging sensitive information, such as passwords or other personal information, to a non-trustworthy source.

recipient: An entity that can receive email messages.

remote operation (ROP): An operation that is invoked against a server. Each ROP represents an action, such as delete, send, or query. A ROP is contained in a ROP buffer for transmission over the wire.

restriction: A filter used to map some domain into a subset of itself, by passing only those items from the domain that match the filter. Restrictions can be used to filter existing Table objects or to define new ones, such as search folder (2) or rule criteria.

ROP request: See ROP request buffer.

rule: An item that defines a condition and an action. The condition is evaluated for each **Message object** as it is delivered, and the action is executed if the new Message object matches the condition.

Simple Mail Transfer Protocol (SMTP): A member of the TCP/IP suite of protocols that is used to transport Internet messages, as described in [RFC5321].

spam: An unsolicited email message.

spam filter: A filter that checks certain conditions in a message to determine a spam confidence level.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined 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.

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

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

[MS-OXOABK] Microsoft Corporation, "Address Book Object Protocol".

[MS-OXOMSG] Microsoft Corporation, "Email Object Protocol".

[MS-OXORULE] Microsoft Corporation, "Email Rules Protocol".

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

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

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

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

1.2.2 Informative References

[MS-ADTS] Microsoft Corporation, "Active Directory Technical Specification".

[MS-AUTHSOD] Microsoft Corporation, "Authentication Services Protocols Overview".

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

[MS-OXCPRPT] Microsoft Corporation, "Property and Stream Object Protocol".

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

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

[MS-OXPROTO] Microsoft Corporation, "Exchange Server Protocols System Overview".

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008, http://rfc-editor.org/rfc/rfc5321.txt

1.3 Overview

The Spam Confidence Level 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 folder** 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.

The **Junk Email rule**, which is an **extended rule**, specifies the client's spam and **phishing** preferences. When an e-mail message is delivered to a server, the server applies the Junk Email rule against the properties of the e-mail message to determine whether to put the message in the **Junk Email folder**.

Clients can use the junk email move stamp to indicate that a 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 Email folder once. If the user has retrieved a message from the Junk Email folder, it will not be reprocessed. Clients can also set this property to populate a **message store** with trusted **Message objects** that are never spam but might look like spam to a spam filter. The RSS Object Protocol, as described in [MS-OXORSS], is a practical example of this method.

1.4 Relationship to Other Protocols

The Spam Confidence Level Protocol relies on the following protocols:

- The Email Rules Protocol, as described in [MS-OXORULE], to create **rules**.

- The Message and Attachment Object Protocol, as described in [MS-OXCMSG], to create and access Message objects.
- The Folder Object Protocol, as described in [MS-OXCFOLD], to access **Folder objects**.
- The Property and Stream Object Protocol, as described in [MS-OXCPRPT], to get and set properties on Message objects and Folder objects.

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [MS-OXPROTO].

1.5 Prerequisites/Preconditions

This protocol assumes that a system is in place to set and retrieve the properties of Message objects and of 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

The Spam Confidence Level Protocol uses the same underlying transport as that used by the Message and Attachment Object Protocol, as specified in [MS-OXCMSG], and the Email Rules Protocol, as specified in [MS-OXORULE].

2.2 Message Syntax

2.2.1 Message Object Properties

The properties persisted on a Message object are listed in sections 2.2.1.1 through 2.2.1.3.

2.2.1.1 PidLidSpamOriginalFolder Property

Type: **PtypBinary** ([MS-OXCDATA] section 2.11.1)

The **PidLidSpamOriginalFolder** property ([MS-OXPROPS] section 2.301) specifies the folder that contained the message before the message was moved into the Junk Email folder. The value of this property is the **entry ID** of the folder.

2.2.1.2 PidNameExchangeJunkEmailMoveStamp Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidNameExchangeJunkEmailMoveStamp** property ([MS-OXPROPS] section 2.458), if present and valid, indicates that either the message was already processed or the message is safe. The value of this property is valid only if it matches the value at index 5 of the **PidTagAdditionalRenEntryIds** property (section 2.2.3.1).

If the **PidNameExchangeJunkEmailMoveStamp** property is not present or if the value of the **PidNameExchangeJunkEmailMoveStamp** property is not valid, the message MUST be processed by the client's spam filter.

2.2.1.3 PidTagContentFilterSpamConfidenceLevel Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagContentFilterSpamConfidenceLevel** property ([MS-OXPROPS] section 2.716) indicates the likelihood that the e-mail message is spam. The value MUST be in the range -1 to 9 (inclusive). The value -1 indicates that the message is not spam, and a value greater than -1 indicates that the message likely is spam. The greater the number, the higher the likelihood that the message is spam, with 9 indicating the highest likelihood. This property SHOULD be set by the server's spam filter before the Junk Email rule is executed.

2.2.2 Junk Email Rule Properties

The properties persisted on the Junk Email rule are listed in sections 2.2.2.1 through 2.2.2.6.

2.2.2.1 PidTagJunkAddRecipientsToSafeSendersList Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagJunkAddRecipientsToSafeSendersList** property ([MS-OXPROPS] section 2.825) MUST be set to either 0 (zero) or 1. The value 1 indicates that the mail recipients are to be added to the safe senders list. The value zero indicates that the mail recipients are not to be added to the safe senders list. The safe senders list is a collection of e-mail addresses that represent senders whose messages are never marked as spam.

2.2.2.2 PidTagJunkIncludeContacts Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagJunkIncludeContacts** property ([MS-OXPROPS] section 2.826) indicates whether e-mail messages from **contacts** can be treated as junk.

If this property is set to 1, the Junk Email rule MUST specify conditions such that e-mail messages from contacts are never treated as junk. If this property is set to 0 (zero), the Junk Email rule MUST specify conditions such that e-mail messages from contacts can be treated as junk. The conditions of the Junk Email rule are specified in the **PidTagExtendedRuleMessageCondition** property ([MS-OXORULE] section 2.2.4.1.10). For details about creating the Junk Email rule, see section 3.1.4.2.

2.2.2.3 PidTagJunkPermanentlyDelete Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagJunkPermanentlyDelete** property ([MS-OXPROPS] section 2.827) indicates whether spam messages can be permanently deleted. If this property is set to 1, messages identified as spam can be permanently deleted. If this property is set to 0 (zero), messages identified as spam cannot be permanently deleted.

2.2.2.4 PidTagJunkPhishingEnableLinks Property

Type: **PtypBoolean** ([MS-OXCDATA] section 2.11.1)

The **PidTagJunkPhishingEnableLinks** property ([MS-OXPROPS] section 2.828) indicates whether the phishing stamp on the message can be ignored. If the value is nonzero (TRUE), the phishing stamp, as specified in [MS-OXPHISH] section 2.2.1.1, can be ignored. If the value is zero (FALSE), the phishing stamp on the message cannot be ignored.

2.2.2.5 PidTagJunkThreshold Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagJunkThreshold** property ([MS-OXPROPS] section 2.829) indicates how aggressively the client is to send incoming mail to the Junk Email folder. When the value is 0xFFFFFFFF, spam filtering SHOULD NOT be applied; however, the blocked sender domains clause of the Junk Email rule 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 valid values for this property.

Value	Meaning
0xFFFFFFFF	No spam filtering
0x00000006	Low spam filtering
0x00000003	High spam filtering

Value	Meaning
0x80000000	Trusted lists only

2.2.2.6 PidTagReportTime Property

Type: **PtypTime** ([MS-OXCDATA] section 2.11.1)

The **PidTagReportTime** property ([MS-OXPROPS] section 2.991) indicates the last time the contact list that is controlled by the **PidTagJunkIncludeContacts** property (section 2.2.2.2) was updated.

2.2.3 Inbox Folder Properties

The property listed in section 2.2.3.1 is on the Inbox folder.

2.2.3.1 PidTagAdditionalRenEntryIds Property

Type: **PtypMultipleBinary** ([MS-OXCDATA] section 2.11.1)

The **PidTagAdditionalRenEntryIds** property ([MS-OXOSFLD] section 2.2.3) is persisted on the Inbox folder of a message store. The value at zero-based index five of this property is used to validate the **PidNameExchangeJunkEmailMoveStamp** property (section 2.2.1.2), as specified in section 3.1.4.1.

2.2.4 Format of the Junk Email Rule

The Junk Email rule stores preferences regarding how spam filtering is applied.

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

The **restriction** that makes up the condition of the Junk Email 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 clauses are listed in the following table.

Blocked sender addresses	E-mail addresses of senders (who the message was sent from) to be blocked
Blocked sender domains	E-mail domains of senders that can be blocked.
Trusted sender domains	E-mail domains of senders that are trusted.
Trusted recipient domains	E-mail domains 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.

Blocked sender addresses	**E-mail addresses of senders (who the message was sent from) to be blocked**
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 (section 2.2.1.3) in the event that this property was applied to the message during delivery.

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

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.

The following common abstract data model (ADM) types are defined in this document:

Mailbox

Messaging Object

3.1.1.1 Per Mailbox

Mailboxes are represented by the **Mailbox** ADM type. The following ADM elements are maintained for each **Mailbox** ADM type:

Mailbox.JunkEmailRule: A server-side extended rule that moves all spam messages to the Junk Email folder.

3.1.1.2 Per Messaging Object

Message objects are represented by the **MessagingObject** ADM type.

3.1.2 Timers

None.

3.1.3 Initialization

When the user first interacts with the mailbox, the server SHOULD create the Junk Email rule in the Inbox folder.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Creating the Junk Email Rule

The Junk Email rule is a server-side extended rule that follows the syntax specified in [MS-OXORULE] section 2.2.4.

The Junk Email rule is represented by a **folder associated information (FAI)** message. The Junk Email rule is created or modified by adding or modifying an FAI message as specified in [MS-OXORULE] section 3.1.4.3.2.

The FAI Message object MUST have properties set as follows.

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

The **PidTagSubject** property ([MS-OXCMSG] section 2.2.1.46) 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 the following two **actions** in the format specified for the **PidTagExtendedRuleMessageActions** property:

- An OP_MOVE action to move the message to the Junk Email folder.
- An OP_TAG action to set the **PidNameExchangeJunkEmailMoveStamp** property (section 2.2.1.2) on the message that is moved to the Junk Email folder.

The **PidTagReportTime** (section 2.2.2.6), **PidTagJunkIncludeContacts** (section 2.2.2.2), and **PidTagJunkThreshold** (section 2.2.2.5) properties are set as specified. The **PidTagExtendedRuleMessageCondition** property ([MS-OXORULE] section 2.2.4.1.10) MUST contain the following restrictions. The formats of the restriction structures are specified in [MS-OXCDATA] section 2.12.1 through [MS-OXCDATA] section 2.12.12. All e-mail addresses MUST be of the SMTP address type.

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 subclause MUST be a **RES_CONTENT** restriction with the **FuzzyLevelLow** field set to FL_FULLSTRING and the **FuzzyLevelHigh** field set to FL_IGNORECASE comparing the value of the **PidTagSenderEmailAddress** property ([MS-OXOMSG] section 2.2.1.41) with a string that contains the e-mail address of a bad sender.
    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 the **PidTagContentFilterSpamConfidenceLevel** property (section 2.2.1.3).
                2. A **RES_PROPERTY** restriction for the **PidTagContentFilterSpamConfidenceLevel** property, 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 subclause MUST be a **RES_CONTENT** restriction with the **FuzzyLevelLow** field set to FL_SUBSTRING and the **FuzzyLevelHigh** field set to FL_IGNORECASE comparing the

value of the **PidTagSenderEmailAddress** property with a string that contains the domain of a bad sender.

   2. A **RES_NOT** restriction with a **RES_OR** restriction that has two subclauses:

      1. A **RES_OR** restriction with zero or more subclauses, one for each trusted sender domain. Each subclause MUST be a **RES_CONTENT** restriction with the **FuzzyLevelLow** field set to FL_SUBSTRING and the **FuzzyLevelHigh** field set to FL_IGNORECASE comparing the value of the **PidTagSenderEmailAddress** property with a string that contains the domain of a trusted sender.

      2. A **RES_SUB** restriction for the **PidTagMessageRecipients** property ([MS-OXCMSG] section 2.2.1.47), with a RES_OR restriction with zero or more subclauses, one for each trusted recipient domain. Each subclause MUST be a **RES_CONTENT** restriction with the **FuzzyLevelLow** field set to FL_SUBSTRING and the **FuzzyLevelHigh** field set to FL_IGNORECASE comparing the value of the **PidTagEmailAddress** property ([MS-OXOABK] section 2.2.3.14) with a string that contains the domain of a trusted recipient.

2. A **RES_NOT** restriction with a **RES_OR** restriction that has three subclauses:

   1. A **RES_OR** restriction with zero or more subclauses, one for each trusted sender e-mail address. Each subclause MUST be a **RES_CONTENT** restriction with the **FuzzyLevelLow** field set to FL_FULLSTRING and the **FuzzyLevelHigh** field set to FL_IGNORECASE comparing the value of the **PidTagSenderEmailAddress** property with a string that contains the e-mail address of a trusted sender.

   2. A **RES_SUB** restriction for the **PidTagMessageRecipients** property, with a **RES_OR** restriction with zero or more subclauses, one for each trusted recipient e-mail address. Each subclause MUST be a **RES_CONTENT** restriction with the **FuzzyLevelLow** field set to FL_FULLSTRING and the **FuzzyLevelHigh** field set to FL_IGNORECASE comparing the value of the **PidTagEmailAddress** property with a string that the e-mail address of a trusted recipient.

   3. A **RES_OR** restriction with zero or more subclauses. Each subclause MUST be a **RES_CONTENT** restriction with the **FuzzyLevelLow** field set to FL_SUBSTRING and the **FuzzyLevelHigh** field set to FL_IGNORECASE comparing the value of the **PidTagSenderEmailAddress** property with a string that contains the e-mail address of a contact from the mailbox's contact list. If the **PidTagJunkIncludeContacts** property (section 2.2.2.2) is set to 0 (zero), this restriction MUST be empty (NULL); if the **PidTagJunkIncludeContacts** property is set to 1, then there SHOULD be one of these restrictions for each trusted contact e-mail address.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.5.1 Executing the Junk Email Rule on a Message

When the server executes the Junk Email 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 ([MS-OXORULE] section 2.2.4.1.10) on the Junk Email rule (as specified in section 3.1.4.2).

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

1. Moves the message to the Junk Email folder.

2. Sets the **PidNameExchangeJunkEmailMoveStamp** property (section 2.2.1.2) on the message.

If the **PidTagExtendedRuleMessageCondition** property on the Junk Email 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.

The following common ADM types are defined in this document:

Mailbox

Messaging Object

3.2.1.1 Per Mailbox

Mailboxes are represented by the **Mailbox** ADM type. The following ADM elements are maintained for each **Mailbox** ADM type:

Mailbox.MessagingObject: An abstract representation of a Message object.

3.2.1.2 Per Messaging Object

Message objects are represented by the **MessagingObject** ADM type. The following ADM elements are maintained for each **MessagingObject** ADM type:

MessagingObject.JunkEMailStamp: A client-generated code that prevents a Message object from being flagged as spam by the server's Junk Email rule.

3.2.2 Timers

None.

3.2.3 Initialization

The client SHOULD create the junk email move stamp on the first interaction of the user with a mailbox that requires it.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Obtaining or Creating the Junk Email Move Stamp

The **PidNameExchangeJunkEmailMoveStamp** property (section 2.2.1.2) is set by the client on every message that is moved by the Junk Email rule or is otherwise trusted content.

The value of the **PidNameExchangeJunkEmailMoveStamp** property is valid only if it matches the value in the **PidTagAdditionalRenEntryIds** property (section 2.2.3.1), as specified in section 3.2.4.1.1.

The client MUST obtain or create the junk email 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 Email Move Stamp

To obtain the junk email move stamp, the client MUST read the **PidTagAdditionalRenEntryIds** property (section 2.2.3.1) 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 (section 2.2.1.2), stored as an unsigned **PtypInteger32** ([MS-OXCDATA] section 2.11.1). The client MUST use this value for the **PidNameExchangeJunkEmailMoveStamp** property when creating the Junk Email 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 Email Move Stamp

If there is no value at zero-based index 5 of the **PidTagAdditionalRenEntryIds** property (section 2.2.3.1) of the Inbox folder, the client MUST generate an arbitrary **PtypInteger32** value ([MS-OXCDATA] section 2.11.1) and store it as an unsigned **PtypInteger32** to the zero-based index 5 of the **PidTagAdditionalRenEntryIds** property of the Inbox folder. For security details, see section 5.1.

The client MUST set the value of the **PidNameExchangeJunkEmailMoveStamp** (section 2.2.1.2) 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 Modifying the Junk Email Rule

To store user preferences regarding how spam filtering occurs for a mailbox, the client modifies the Junk Email rule created on the server. The rule itself is executed only on the server.

Clients interpret properties on the message and data in the **PidTagExtendedRuleMessageCondition** property ([MS-OXORULE] section 2.2.4.1.10) of the Junk Email 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** property (section 2.2.1.3) that can be used to determine 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** property, as specified in section 2.2.4. The client MUST format the Junk Email 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 Email 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 Email rule, as specified in [MS-OXORULE] section 3.1.4.3.2, to match these new preferences.

3.2.4.5 Server Junk Email Rule Changes

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

3.2.4.6 User Adds a New Contact to Their Contacts Folder

If the **PidTagJunkIncludeContacts** property (section 2.2.2.2) is present with a value of 1, 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 Email rule. If the added contact's e-mail addresses are already included in the trusted contacts section of the Junk Email 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 Email rule, the client MUST update the server Junk Email 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 the value of the **PidTagJunkIncludeContacts** property is 0 (zero), no action is required.

3.2.4.7 User Sends an E-Mail

If the **PidTagJunkAddRecipientsToSafeSendersList** property (section 2.2.2.1) is present with a value of 1, the client MUST update the server Junk Email 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 Email rule condition.

If the value of the **PidTagJunkAddRecipientsToSafeSendersList** property is 0 (zero), no action is required.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Receiving an E-Mail Message

If the client receives an e-mail message that has the **PidNameExchangeJunkEmailMoveStamp** property (section 2.2.1.2) set by another client, that property MUST be validated against the **PidTagAdditionalRenEntryIds** property (section 2.2.3.1), as specified in section 3.2.4.1.2. If the value matches, the client MUST NOT run a spam filter against the e-mail message. Validating the **PidNameExchangeJunkEmailMoveStamp** property ensures that malicious messaging applications cannot easily circumvent a client's spam filters.

If the client runs a spam filter to determine whether the e-mail message is spam, the client SHOULD use the preferences specified in the Junk Email rule to control the spam filter.

If the client spam filter determines that the e-mail message is spam, the client uses the value of the **PidTagJunkPermanentlyDelete** property (section 2.2.2.3) on the Junk Email rule to determine whether to permanently delete the e-mail message. The client SHOULD set the **PidLidSpamOriginalFolder** property (section 2.2.1.1) on each message that is moved to the Junk Email folder.

The client can use the **PidTagJunkPhishingEnableLinks** property (section 2.2.2.4) on the Junk Email rule to determine whether to enable links within the 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 Email 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 opens the Junk Email rule by using the **RopOpenMessage** **ROP** ([MS-OXCROPS] section 2.2.6.1).

The client retrieves the **PidTagExtendedRuleMessageCondition** property ([MS-OXPROPS] section 2.771) of the Junk Email rule by using the **RopGetPropertiesSpecific** ROP ([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 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
Trusted sender addresses	L"safe@example.com"

List	C-style string representation
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 ROPs:

- **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** ROP 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"

List	C-style string representation
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** **ROP request** ([MS-OXCROPS] section 2.2.6.3) to persist the object on the server and a **RopRelease** ROP request to release the object.

5 Security

5.1 Security Considerations for Implementers

The **PidNameExchangeJunkEmailMoveStamp** property (section 2.2.1.2) is used to bypass content protection offered by client spam filters. If the valid junk email 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 the procedure in section 3.2.4.1.2 in such a way that the value of the zero-based index 5 of the **PidTagAdditionalRenEntryIds** property (section 2.2.3.1) 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 Exchange Server 2013
- Microsoft Office Outlook 2003
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016 Preview

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.

7 Change Tracking

This section identifies changes that were made to this document since the last release. 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.
- The removal of a document from the documentation set.

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 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 changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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.
- 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 dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
6 Appendix A: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.

8 Index