

[MS-OXWSRULES]:

Inbox Rules Web Service Protocol

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 [Open Specification Promise](#) or the [Community Promise](#). 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. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **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.

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional

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

Preliminary

Revision Summary

Date	Revision History	Revision Class	Comments
5/5/2010	0.1.0	Major	Initial Release.
8/4/2010	1.0	Major	Significantly changed the technical content.
11/3/2010	2.0	Major	Significantly changed the technical content.
3/18/2011	3.0	Major	Significantly changed the technical content.
8/5/2011	4.0	Major	Significantly changed the technical content.
10/7/2011	4.1	Minor	Clarified the meaning of the technical content.
1/20/2012	5.0	Major	Significantly changed the technical content.
4/27/2012	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	5.1	Minor	Clarified the meaning 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	5.4	Minor	Clarified the meaning of the technical content.
2/10/2014	5.4	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	5.5	Minor	Clarified the meaning of the technical content.
7/31/2014	5.6	Minor	Clarified the meaning of the technical content.
10/30/2014	5.7	Minor	Clarified the meaning of the technical content.
5/26/2015	6.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	8
1.2.1	Normative References	8
1.2.2	Informative References	9
1.3	Overview	9
1.4	Relationship to Other Protocols	9
1.5	Prerequisites/Preconditions	9
1.6	Applicability Statement	10
1.7	Versioning and Capability Negotiation	10
1.8	Vendor-Extensible Fields	10
1.9	Standards Assignments.....	10
2	Messages.....	11
2.1	Transport.....	11
2.2	Common Message Syntax	11
2.2.1	Namespaces	11
2.2.2	Messages.....	11
2.2.3	Elements	11
2.2.4	Complex Types.....	12
2.2.4.1	t:RuleActionsType Complex Type	12
2.2.4.2	t:RulePredicateDateRangeType Complex Type	14
2.2.4.3	t:RulePredicateSizeRangeType Complex Type	15
2.2.4.4	t:RulePredicatesType Complex Type	15
2.2.4.5	t:RuleType Complex Type	21
2.2.5	Simple Types	23
2.2.5.1	t:FlaggedForActionType Simple Type	23
2.2.6	Attributes	24
2.2.7	Groups	24
2.2.8	Attribute Groups.....	24
3	Protocol Details.....	25
3.1	ExchangeServicePortType Server Detail	25
3.1.1	Abstract Data Model.....	25
3.1.2	Timers	25
3.1.3	Initialization.....	25
3.1.4	Message Processing Events and Sequencing Rules	25
3.1.4.1	GetInboxRules.....	25
3.1.4.1.1	Messages	26
3.1.4.1.1.1	tns:GetInboxRulesSoapIn Message	26
3.1.4.1.1.2	tns:GetInboxRulesSoapOut Message	27
3.1.4.1.2	Elements.....	27
3.1.4.1.2.1	GetInboxRules Element	27
3.1.4.1.2.2	GetInboxRulesResponse Element	28
3.1.4.1.3	Complex Types	28
3.1.4.1.3.1	m:GetInboxRulesRequestType Complex Type	28
3.1.4.1.3.2	m:GetInboxRulesResponseType Complex Type	29
3.1.4.1.3.3	t:ArrayOfRulesType Complex Type	29
3.1.4.1.4	Simple Types	30
3.1.4.1.5	Attributes	30
3.1.4.1.6	Groups.....	30
3.1.4.1.7	Attribute Groups.....	30
3.1.4.2	UpdateInboxRules.....	30
3.1.4.2.1	Messages	31

3.1.4.2.1.1	tns:UpdateInboxRulesSoapIn Message	31
3.1.4.2.1.2	tns:UpdateInboxRulesSoapOut Message	32
3.1.4.2.2	Elements	32
3.1.4.2.2.1	UpdateInboxRules Element	32
3.1.4.2.2.2	UpdateInboxRulesResponse Element	33
3.1.4.2.3	Complex Types	33
3.1.4.2.3.1	m:UpdateInboxRulesRequestType Complex Type	33
3.1.4.2.3.2	m:UpdateInboxRulesResponseType Complex Type	34
3.1.4.2.3.3	t:ArrayOfRuleOperationsType Complex Type	35
3.1.4.2.3.4	t:RuleOperationType Complex Type	35
3.1.4.2.3.5	t:CreateRuleOperationType Complex Type	36
3.1.4.2.3.6	t:SetRuleOperationType Complex Type	36
3.1.4.2.3.7	t>DeleteRuleOperationType Complex Type	37
3.1.4.2.3.8	t:ArrayOfRuleOperationErrorsType Complex Type	37
3.1.4.2.3.9	t:RuleOperationErrorType Complex Type	38
3.1.4.2.3.10	t:ArrayOfRuleValidationErrorsType Complex Type	38
3.1.4.2.3.11	t:RuleValidationErrorType Complex Type	39
3.1.4.2.4	Simple Types	40
3.1.4.2.4.1	t:RuleFieldURIType Simple Type	40
3.1.4.2.4.2	t:RuleValidationErrorCodeType Simple Type	47
3.1.4.2.5	Attributes	49
3.1.4.2.6	Groups	50
3.1.4.2.7	Attribute Groups	50
3.1.5	Timer Events	50
3.1.6	Other Local Events	50
4	Protocol Examples	51
4.1	CreateInboxRule	51
4.2	SetInboxRule	52
4.3	DeleteInboxRule	53
4.4	GetInboxRules	54
4.5	RuleOperationErrorType	55
4.6	RuleValidationErrorType	56
5	Security	59
5.1	Security Considerations for Implementers	59
5.2	Index of Security Parameters	59
6	Appendix A: Full WSDL	60
7	Appendix B: Full XML Schema	62
7.1	Messages Schema	62
7.2	Types Schema	63
8	Appendix C: Product Behavior	69
9	Change Tracking	70
10	Index	72

1 Introduction

The Inbox Rules Web Service Protocol is used to get Inbox **rules** and update Inbox rules for messages on 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 (2)** are TRUE. A rule contains one or more actions.

binary large object (BLOB): A discrete packet of data that is stored in a database and is treated as a sequence of uninterpreted bytes.

client-side rule: A **rule** that has at least one **action** that is executed by a client because it cannot be executed by a server.

Deleted Items folder: A special folder that is the default location for objects that have been deleted.

email address: A string that identifies a user and enables the user to receive Internet messages.

endpoint: A communication port that is exposed by an application server for a specific shared service and to which messages can be addressed.

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS): An extension of **HTTP** that securely encrypts and decrypts webpage requests.

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.

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

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

non-delivery report: A report message that is generated and sent by a server to the sender of a message if an email message could not be received by an intended recipient.

rule: (1) A condition or action, or a set of conditions or actions, that performs tasks automatically based on events and values.

(2) 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.

S/MIME (Secure/Multipurpose Internet Mail Extensions): A set of cryptographic security services, as described in [\[RFC5751\]](#).

server-side rule: A **rule** for which all actions are executed by a server.

Short Message Service (SMS): A communications protocol that is designed for sending text messages between mobile phones.

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

SOAP action: The HTTP request header field used to indicate the intent of the SOAP request, using a **URI** value. See [\[SOAP1.1\]](#) section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a **SOAP message** to its recipient. See [\[SOAP1.2-1/2007\]](#) section 5.3 for more information.

SOAP header: A mechanism for implementing extensions to a **SOAP message** in a decentralized manner without prior agreement between the communicating parties. See [\[SOAP1.2-1/2007\]](#) section 5.2 for more information.

SOAP message: An **XML** document consisting of a mandatory SOAP envelope, an optional **SOAP header**, and a mandatory **SOAP body**. See [\[SOAP1.2-1/2007\]](#) section 5 for more information.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [\[RFC3986\]](#).

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [\[RFC1738\]](#).

web server: A server computer that hosts websites and responds to requests from applications.

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

WSDL message: An abstract, typed definition of the data that is communicated during a WSDL operation [\[WSDL\]](#). Also, an element that describes the data being exchanged between web service providers and clients.

WSDL port type: A named set of logically-related, abstract **Web Services Description Language (WSDL)** operations and messages.

XML: The Extensible Markup Language, as described in [\[XML1.0\]](#).

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [\[RFC3986\]](#). A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [\[XMLNS-2ED\]](#).

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [\[XML\]](#).

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic

syntax constraints that are imposed by **XML** itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).

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-OXWSCDATA] Microsoft Corporation, "[Common Web Service Data Types](#)".

[MS-OXWSCORE] Microsoft Corporation, "[Core Items Web Service Protocol](#)".

[MS-OXWSFOLD] Microsoft Corporation, "[Folders and Folder Permissions Web Service Protocol](#)".

[MS-OXWSGTZ] Microsoft Corporation, "[Get Server Time Zone Web Service Protocol](#)".

[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>

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.rfc-editor.org/rfc/rfc2616.txt>

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <http://www.rfc-editor.org/rfc/rfc2818.txt>

[RFC3066] Alvestrand, H., "Tags for the Identification of Languages", BCP 47, RFC 3066, January 2001, <http://www.ietf.org/rfc/rfc3066.txt>

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

1.2.2 Informative References

[MS-OXDCLI] Microsoft Corporation, "[Autodiscover Publishing and Lookup Protocol](#)".

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

[MS-OXWSADISC] Microsoft Corporation, "[Autodiscover Publishing and Lookup SOAP-Based Web Service Protocol](#)".

1.3 Overview

This protocol enables clients to get and update Inbox rules (2), which are sets of conditions and associated **actions**. Clients use Inbox rules (2) to automatically organize, categorize, and act on messages as the messages are delivered to a folder.

Server use rules (2) on mailboxes and other folders to automatically execute actions on objects in the folders. Rules (2) are used to carry out predefined or custom actions. Rules (2) can be performed on the server **message store (server-side rule)** or on the client (**client-side rule**).

1.4 Relationship to Other Protocols

A client that implements this protocol can use the Autodiscover Publishing and Lookup SOAP-Based Web Service Protocol, as described in [\[MS-OXWSADISC\]](#), or the Autodiscover Publishing and Lookup Protocol, as described in [\[MS-OXDCLI\]](#), to identify the target **endpoint** to use for each operation.

This protocol uses the SOAP Protocol, as described in [\[SOAP1.1\]](#), to specify the structure information exchanged between the client and server. This protocol uses the XML Protocol, as described in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), to describe the message content sent to and from the server.

This protocol uses SOAP over **HTTP**, as described in [\[RFC2616\]](#), and SOAP over **HTTPS**, as described in [\[RFC2818\]](#), as shown in the following figure.

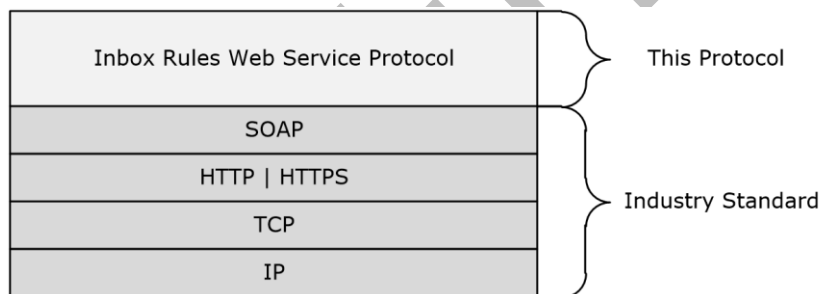


Figure 1: This protocol in relation to other protocols

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

1.5 Prerequisites/Preconditions

The endpoint **URL** that is returned by either the Autodiscover Publishing and Lookup SOAP-Based Web Service Protocol, as described in [\[MS-OXWSADISC\]](#), or the Autodiscover Publishing and Lookup Protocol, as described in [\[MS-OXDCLI\]](#), is required to form the HTTP request to the **web server** that hosts this protocol. The operations that this protocol defines cannot be accessed unless the correct endpoint is identified in the HTTP web requests that target this protocol.

1.6 Applicability Statement

This protocol is applicable to environments that get or update the rules (2) that are applied to a user's **mailbox**. This web service protocol is applicable to SOAP-based clients.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with SOAP 1.1, as specified in section [2.1](#).
- **Protocol Versions:** This protocol has only one **WSDL port type** version. The **WSDL** version of the request is identified by using the **t:RequestServerVersion** element, as described in [\[MS-OXWSCDATA\]](#) section 2.2.3.11, and the version of the server responding to the request is identified using the **t:ServerVersionInfo** element, as described in [\[MS-OXWSCDATA\]](#) section 2.2.3.12.
- **Security and Authentication Methods:** This protocol relies on the web server that is hosting it to perform authentication.
- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section [3.1.4](#).
- **Capability Negotiation:** This protocol does not support version negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The WSDL in this specification provides a base description of the protocol. The schema in this specification provides a base description of the message syntax. The text that specifies the WSDL and schema might specify restrictions that reflect actual protocol behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, or **present**.

2.1 Transport

The SOAP version supported is SOAP 1.1. For details, see [\[SOAP1.1\]](#).

This protocol relies on the web server that hosts the application to perform authentication. The protocol **MUST** support SOAP over HTTP, as specified in [\[RFC2616\]](#). The protocol **SHOULD** use secure communications via HTTPS, as specified in [\[RFC2818\]](#).

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema**, as defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and Web Services Description Language (WSDL), as defined in [\[WSDL\]](#).

2.2.1 Namespaces

This specification defines and references various **XML namespaces** by using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific **XML namespace prefix** for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/exchange/services/2006/messages	
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
m	http://schemas.microsoft.com/exchange/services/2006/messages	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	

2.2.2 Messages

This specification does not define any common **WSDL message** definitions.

2.2.3 Elements

This specification does not define any common XML schema element definitions.

2.2.4 Complex Types

The following table summarizes the set of common XML schema complex type definitions that are defined by this specification. XML schema complex type definitions that are specific to a particular operation are defined with the operation.

Complex type name	Description
RuleActionsType	Specifies the set of actions that are available to a rule (2).
RulePredicateDateRangeType	Specifies the date range in the RulePredicatesType complex type object.
RulePredicateSizeRangeType	Specifies the size range in the RulePredicatesType complex type object.
RulePredicatesType	Specifies the set of conditions and exceptions that are available for a rule (2).
RuleType	Represents a rule (2) in the user's mailbox.

2.2.4.1 t:RuleActionsType Complex Type

The **RuleActionsType** complex type specifies the set of actions that are available to a rule (2).

```
<xs:complexType name="RuleActionsType">
  <xs:sequence>
    <xs:element name="AssignCategories"
      type="t:ArrayOfStringsType"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="CopyToFolder"
      type="t:TargetFolderIdType"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="Delete"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="ForwardAsAttachmentToRecipients"
      type="t:ArrayOfEmailAddressesType"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="ForwardToRecipients"
      type="t:ArrayOfEmailAddressesType"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="MarkImportance"
      type="t:ImportanceChoicesType"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="MarkAsRead"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="MoveToFolder"
      type="t:TargetFolderIdType"
    />
  </xs:sequence>
</xs:complexType>
```

```

        minOccurs="0"
        maxOccurs="1"
    />
    <xs:element name="PermanentDelete"
        type="xs:boolean"
        minOccurs="0"
        maxOccurs="1"
    />
    <xs:element name="RedirectToRecipients"
        type="t:ArrayOfEmailAddressesType"
        minOccurs="0"
        maxOccurs="1"
    />
    <xs:element name="SendSMSAlertToRecipients"
        type="t:ArrayOfEmailAddressesType"
        minOccurs="0"
        maxOccurs="1"
    />
    <xs:element name="ServerReplyWithMessage"
        type="t:ItemIdType"
        minOccurs="0"
        maxOccurs="1"
    />
    <xs:element name="StopProcessingRules"
        type="xs:boolean"
        minOccurs="0"
        maxOccurs="1"
    />
</xs:sequence>
</xs:complexType>

```

The following table lists the child elements of the **RuleActionsType** complex type.

Element name	Type	Description
AssignCategories	t:ArrayOfStringsType ([MS-OXWSCDATA] section 2.2.4.13)	The categories that are stamped on email messages.
CopyToFolder	t:TargetFolderIdType ([MS-OXWSFOLD] section 2.2.4.14)	The ID of the folder that email items will be copied to.
Delete	xs:boolean ([XMLSCHEMA2])	A value that indicates whether messages are to be moved to the Deleted Items folder .
ForwardAsAttachmentToRecipients	t:ArrayOfEmailAddressesType ([MS-OXWSCDATA] section 2.2.4.7)	The email addresses to which messages are to be forwarded as attachments.
ForwardToRecipients	t:ArrayOfEmailAddressesType	The email addresses to which messages are to be forwarded.
MarkImportance	t:ImportanceChoicesType ([MS-OXWSCORE] section 2.2.5.3)	The importance that is to be stamped on messages.
MarkAsRead	xs:boolean	A value that indicates whether messages are to be marked as read.
MoveToFolder	t:TargetFolderIdType	The ID of the folder that

Element name	Type	Description
		email items will be moved to.
PermanentDelete	xs:boolean	A value that indicates whether messages are to be permanently deleted and not saved to the Deleted Items folder.
RedirectToRecipients	t:ArrayOfEmailAddressesType	The email addresses to which messages are to be redirected.
SendSMSAlertToRecipients	t:ArrayOfEmailAddressesType	The mobile phone numbers to which a Short Message Service (SMS) alert is to be sent.
ServerReplyWithMessage	t:ItemIdType ([MS-OXWSCORE] section 2.2.4.24)	The ID of the template message that is to be sent as a reply to incoming messages.
StopProcessingRules	xs:boolean	A value that indicates whether subsequent rules (2) are to be evaluated.

2.2.4.2 t:RulePredicateDateRangeType Complex Type

The **RulePredicateDateRangeType** complex type specifies the date range in the **RulePredicatesType** complex type.

```
<xs:complexType name="RulePredicateDateRangeType">
  <xs:sequence>
    <xs:element name="StartDateTime"
      type="xs:dateTime"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="EndDateTime"
      type="xs:dateTime"
      minOccurs="0"
      maxOccurs="1"
    />
  </xs:sequence>
</xs:complexType>
```

The following table lists the child elements of the **RulePredicateDateRangeType** complex type.

Element name	Type	Description
StartDateTime	xs:dateTime (XMLSCHEMA21)	Specifies the rule (2) time period and indicates that the rule (2) condition is met after this value.
EndDateTime	xs:dateTime	Specifies the rule (2) time period and indicates that the rule (2) condition is met before this value.

This complex type MUST contain a **StartDateTime** or an **EndDateTime** element and can contain both a **StartDateTime** and an **EndDateTime** element.

The time period specified by the **StartDateTime** element MUST NOT be after the time period specified by the **EndDateTime** element.

2.2.4.3 t:RulePredicateSizeRangeType Complex Type

The **RulePredicateSizeRangeType** complex type specifies the size range in the **RulePredicatesType** complex type. This complex type represents the minimum and maximum size of a message in the rule (2) predicate, in kilobytes.

```
<xs:complexType name="RulePredicateSizeRangeType">
  <xs:sequence>
    <xs:element name="MinimumSize"
      type="xs:int"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="MaximumSize"
      type="xs:int"
      minOccurs="0"
      maxOccurs="1"
    />
  </xs:sequence>
</xs:complexType>
```

The following table lists the child elements of the **RulePredicateSizeRangeType** complex type.

Element name	Type	Description
MinimumSize	xs:int (XMLSCHEMA21)	The minimum size of a message for the conditions or exception to apply.
MaximumSize	xs:int	The maximum size of a message for the conditions or exception to apply.

This complex type can contain a **MinimumSize** and a **MaximumSize** element.

The value of the **MinimumSize** element MUST NOT be greater than value of the **MaximumSize** element. The values of the **MinimumSize** and **MaximumSize** elements MUST NOT be negative.

To disable this predicate, set both the **MinimumSize** and the **MaximumSize** element to null.

2.2.4.4 t:RulePredicatesType Complex Type

The **RulePredicatesType** complex type specifies the set of conditions and exceptions that are available for a rule (2).

```
<xs:complexType name="RulePredicatesType">
  <xs:sequence>
    <xs:element name="Categories"
      type="t:ArrayOfStringsType"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="ContainsBodyStrings"

```

```

    type="t:ArrayOfStringsType"
    minOccurs="0"
    maxOccurs="1"
  />
<xs:element name="ContainsHeaderStrings"
  type="t:ArrayOfStringsType"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="ContainsRecipientStrings"
  type="t:ArrayOfStringsType"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="ContainsSenderStrings"
  type="t:ArrayOfStringsType"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="ContainsSubjectOrBodyStrings"
  type="t:ArrayOfStringsType"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="ContainsSubjectStrings"
  type="t:ArrayOfStringsType"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="FlaggedForAction"
  type="t:FlaggedForActionType"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="FromAddresses"
  type="t:ArrayOfEmailAddressesType"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="FromConnectedAccounts"
  type="t:ArrayOfStringsType"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="HasAttachments"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="Importance"
  type="t:ImportanceChoicesType"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="IsApprovalRequest"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="IsAutomaticForward"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
  />
<xs:element name="IsAutomaticReply"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"

```



```

/>
<xs:element name="IsEncrypted"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="IsMeetingRequest"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="IsMeetingResponse"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="IsNDR"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="IsPermissionControlled"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="IsReadReceipt"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="IsSigned"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="IsVoicemail"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="ItemClasses"
  type="t:ArrayOfStringsType"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="MessageClassifications"
  type="t:ArrayOfStringsType"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="NotSentToMe"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="SentCcMe"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="SentOnlyToMe"
  type="xs:boolean"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="SentToAddresses"
  type="t:ArrayOfEmailAddressesType"

```

```

    minOccurs="0"
    maxOccurs="1"
  />
  <xs:element name="SentToMe"
    type="xs:boolean"
    minOccurs="0"
    maxOccurs="1"
  />
  <xs:element name="SentToOrCcMe"
    type="xs:boolean"
    minOccurs="0"
    maxOccurs="1"
  />
  <xs:element name="Sensitivity"
    type="t:SensitivityChoicesType"
    minOccurs="0"
    maxOccurs="1"
  />
  <xs:element name="WithinDateRange"
    type="t:RulePredicateDateRangeType"
    minOccurs="0"
    maxOccurs="1"
  />
  <xs:element name="WithinSizeRange"
    type="t:RulePredicateSizeRangeType"
    minOccurs="0"
    maxOccurs="1"
  />
</xs:sequence>
</xs:complexType>

```

The following table lists the child elements of the **RulePredicatesType** complex type.

Element name	Type	Description
Categories	t:ArrayOfStringsType ([MS-OXWSCDATA] section 2.2.4.13)	The categories that an incoming message is labeled with in order for the condition or exception to apply.
ContainsBodyStrings	t:ArrayOfStringsType	The strings that appear in the body of incoming messages in order for the condition or exception to apply.
ContainsHeaderStrings	t:ArrayOfStringsType	The strings that appear in the headers of incoming messages in order for the condition or exception to apply.
ContainsRecipientStrings	t:ArrayOfStringsType	The strings that appear in either the ToRecipients or CcRecipients properties of incoming messages in order for the condition or exception to apply. The ToRecipients property specifies a collection of recipients of an email. The CcRecipients property specifies a collection of recipients that receive a carbon copy (Cc) of an email.
ContainsSenderStrings	t:ArrayOfStringsType	The strings that appear in the From property of incoming messages in order for the condition or exception

Element name	Type	Description
		to apply. The From property specifies the addressee from whom the message was sent.
ContainsSubjectOrBodyStrings	t:ArrayOfStringsType	The strings that appear in either the body or the subject of incoming messages in order for the condition or exception to apply.
ContainsSubjectStrings	t:ArrayOfStringsType	The strings that appear in the subject of incoming messages in order for the condition or exception to apply.
FlaggedForAction	t:FlaggedForActionType (section 2.2.5.1)	The flag for action value that appears on incoming messages in order for the condition or exception to apply.
FromAddresses	t:ArrayOfEmailAddressesType ([MS-OXWSCDATA] section 2.2.4.7)	The email addresses of the senders of incoming messages that appear in order for the condition or exception to apply.
FromConnectedAccounts	t:ArrayOfStringsType	The email account names from which incoming messages have to have been aggregated in order for the condition or exception to apply.
HasAttachments	xs:boolean ([XMLSCHEMA2])	A value that indicates whether incoming messages have to have attachments in order for the condition or exception to apply.
Importance	t:ImportanceChoicesType ([MS-OXWSCORE] section 2.2.5.3)	The importance that is stamped on incoming messages in order for the condition or exception to apply.
IsApprovalRequest	xs:boolean	A value that indicates whether incoming messages have to be approval requests in order for the condition or exception to apply.
IsAutomaticForward	xs:boolean	A value that indicates whether incoming messages have to be automatic forwards in order for the condition or exception to apply.
IsAutomaticReply	xs:boolean	A value that indicates whether incoming messages have to be automatic replies in order for the condition or exception to apply.
IsEncrypted	xs:boolean	A value that indicates whether incoming messages have to be S/MIME encrypted in order for the condition or exception to apply.
IsMeetingRequest	xs:boolean	A value that indicates whether incoming messages have to be meeting requests in order for the

Element name	Type	Description
		condition or exception to apply.
IsMeetingResponse	xs:boolean	A value that indicates whether incoming messages have to be meeting responses in order for the condition or exception to apply.
IsNDR	xs:boolean	A value that indicates whether incoming messages have to be non-delivery reports (NDRs) in order for the condition or exception to apply.
IsPermissionControlled	xs:boolean	A value that indicates whether incoming messages have to be permission controlled (RMS protected) in order for the condition or exception to apply.
IsReadReceipt	xs:boolean	A value that indicates whether incoming messages have to be read receipts in order for the condition or exception to apply.
IsSigned	xs:boolean	A value that indicates whether incoming messages have to be S/MIME signed in order for the condition or exception to apply.
IsVoicemail	xs:boolean	A value that indicates whether incoming messages have to be voice mails in order for the condition or exception to apply.
ItemClasses	t:ArrayOfStringsType	The item classes that have to be stamped on incoming messages in order for the condition or exception to apply.
MessageClassifications	t:ArrayOfStringsType	The message classifications that have to be stamped on incoming messages in order for the condition or exception to apply.
NotSentToMe	xs:boolean	A value that indicates whether the owner of the mailbox cannot be in the ToRecipients property of the incoming messages in order for the condition or exception to apply.
SentCcMe	xs:boolean	A value that indicates whether the owner of the mailbox has to be in the CcRecipients property of incoming messages in order for the condition or exception to apply.
SentOnlyToMe	xs:boolean	A value that indicates whether the owner of the mailbox has to be the only one in the ToRecipients property of incoming messages in order for the condition or exception

Element name	Type	Description
		to apply.
SentToAddresses	t:ArrayOfEmailAddressesType	The email addresses that incoming messages have to have been sent to in order for the condition or exception to apply.
SentToMe	xs:boolean	A value that indicates whether the owner of the mailbox has to be in the ToRecipients property of incoming messages in order for the condition or exception to apply.
SentToOrCcMe	xs:boolean	A value that indicates whether the owner of the mailbox has to be in either a ToRecipients or CcRecipients property of incoming messages in order for the condition or exception to apply.
Sensitivity	t:SensitivityChoicesType ([MS-OXWSCDATA] section 2.2.5.25)	The sensitivity that has to be stamped on incoming messages in order for the condition or exception to apply.
WithinDateRange	t:RulePredicateDateRangeType (section 2.2.4.2)	The date range within which incoming messages have to have been received in order for the condition or exception to apply.
WithinSizeRange	t:RulePredicateSizeRangeType (section 2.2.4.3)	The minimum and maximum sizes that incoming messages have to have in order for the condition or exception to apply.

2.2.4.5 t:RuleType Complex Type

The **RuleType** complex type represents a rule (2) in the user's mailbox.

```
<xs:complexType name="RuleType">
  <xs:sequence>
    <xs:element name="RuleId"
      type="xs:string"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="DisplayName"
      type="xs:string"
      maxOccurs="1"
      minOccurs="1"
    />
    <xs:element name="Priority"
      type="xs:int"
      minOccurs="1"
      maxOccurs="1"
    />
    <xs:element name="IsEnabled"
      type="xs:boolean"
      maxOccurs="1"
    />
  </xs:sequence>
</xs:complexType>
```

```

        minOccurs="1"
      />
    <xs:element name="IsNotSupported"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="IsInError"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="Conditions"
      type="t:RulePredicatesType"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="Exceptions"
      type="t:RulePredicatesType"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="Actions"
      type="t:RuleActionsType"
      minOccurs="0"
      maxOccurs="1"
    />
  </xs:sequence>
</xs:complexType>

```

The following table lists the child elements of the **RuleType** complex type.

Element	Type	Description
RuleId	xs:string ([XMLSCHEMA2])	The rule (2) identifier.
DisplayName	xs:string	A value that contains the display name of a rule (2).
Priority	xs:int ([XMLSCHEMA2])	A value that indicates the order in which a rule (2) is to be run.
IsEnabled	xs:boolean ([XMLSCHEMA2])	Indicates whether the rule (2) is enabled.
IsNotSupported	xs:boolean	Indicates whether the rule (2) cannot be modified with the managed code APIs.
IsInError	xs:boolean	Indicates whether the rule (2) is in an error condition.
Conditions	t:RulePredicatesType (section 2.2.4.4)	The conditions that, when fulfilled, will trigger the rule (2) actions for that rule (2).
Exceptions	t:RulePredicatesType (section 2.2.4.4)	The exceptions that represent all the available rule (2) exception conditions for the inbox rule (2).
Actions	t:RuleActionsType (section 2.2.4.1)	The actions to be taken on a message when the conditions are fulfilled.

This complex type contains all required fields to represent an Inbox rule (2).

The **RuleType** complex type can be extended to represent other rule (2) types.

The **CreateRuleOperationType** complex type, as specified in section [3.1.4.2.3.5](#), does not include a **RuleId** element in the **RuleType** complex type. The rule ID will be generated when the rule (2) is successfully created in the server message store.

2.2.5 Simple Types

The following table summarizes the XML schema simple types that are specific to this operation.

Simple type name	Description
FlaggedForActionType	Defines the follow-up actions that can be stamped on a message.

2.2.5.1 t:FlaggedForActionType Simple Type

The **FlaggedForActionType** simple type defines the follow-up actions that can be stamped on a message.

```
<xs:simpleType name="FlaggedForActionType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="Any"
    />
    <xs:enumeration
      value="Call"
    />
    <xs:enumeration
      value="DoNotForward"
    />
    <xs:enumeration
      value="FollowUp"
    />
    <xs:enumeration
      value="FYI"
    />
    <xs:enumeration
      value="Forward"
    />
    <xs:enumeration
      value="NoResponseNecessary"
    />
    <xs:enumeration
      value="Read"
    />
    <xs:enumeration
      value="Reply"
    />
    <xs:enumeration
      value="ReplyToAll"
    />
    <xs:enumeration
      value="Review"
    />
  </xs:restriction>
</xs:simpleType>
```

The following table lists the values that are defined by the **FlaggedForActionType** simple type.

Value	Meaning
Any	The message is flagged with any action.
Call	The recipient is requested to call the sender.
DoNotForward	The recipient is requested not to forward the message.
FollowUp	The recipient is requested to follow up on the message.
FYI	The recipient received the message for information.
Forward	The recipient is requested to forward the message.
NoResponseNecessary	The recipient is informed that a response to the message is not required.
Read	The recipient is requested to read the message.
Reply	The recipient is requested to reply to the sender of the message.
ReplyToAll	The recipient is requested to reply to everyone the message was sent to.
Review	The recipient is requested to review the message.

2.2.6 Attributes

This specification does not define any common XML schema attribute definitions.

2.2.7 Groups

This specification does not define any common XML schema group definitions.

2.2.8 Attribute Groups

This specification does not define any common XML schema attribute group definitions.

3 Protocol Details

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

3.1 ExchangeServicePortType Server Detail

The Inbox Rules Web Service Protocol defines a single port type with two operations. The operations enable client implementations to get, create, set, and delete rules (2).

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

The following table summarizes the list of WSDL operations as defined by this specification.

Operation name	Description
GetInboxRules	Retrieves a collection of Inbox rules (2) in the identified user's mailbox.
UpdateInboxRules	Updates the authenticated user's Inbox rules (2) in the server message store. This operation enables the user to create, set, and delete Inbox rules (2).

3.1.4.1 GetInboxRules

The **GetInboxRules** operation retrieves Inbox rules (2) in the identified user's mailbox.

The following is the WSDL port type specification of the operation.

```
<wsdl:operation name="GetInboxRules">
  <wsdl:input message="tns:GetInboxRulesSoapIn" />
  <wsdl:output message="tns:GetInboxRulesSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification of the operation.

```
<wsdl:operation name="GetInboxRules">
  <soap:operation
    soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetInboxRules" />
```

```

<wsdl:input>
  <soap:header message="tns:GetInboxRulesSoapIn" part="Impersonation" use="literal" />
  <soap:header message="tns:GetInboxRulesSoapIn" part="MailboxCulture" use="literal" />
  <soap:header message="tns:GetInboxRulesSoapIn" part="RequestVersion" use="literal" />
  <soap:header message="tns:GetInboxRulesSoapIn" part="TimeZoneContext" use="literal"
/>
  <soap:body parts="request" use="literal" />
</wsdl:input>
<wsdl:output>
  <soap:body parts="GetInboxRulesResult" use="literal" />
  <soap:header message="tns:GetInboxRulesSoapOut" part="ServerVersion" use="literal" />
</wsdl:output>
</wsdl:operation>

```

The protocol client sends a **GetInboxRulesSoapIn** request WSDL message and the protocol server responds with a **GetInboxRulesSoapOut** response WSDL message.

A successful **GetInboxRules** operation request returns a **GetInboxRulesResponse** element with the **ResponseClass** attribute of the **GetInboxRulesResponseType** element set to "Success". The **ResponseCode** element of the **GetInboxRulesResponse** element is set to "NoError".

3.1.4.1.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message name	Description
GetInboxRulesSoapIn	Specifies the SOAP message that gets the Inbox rules (2).
GetInboxRulesSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.1.1.1 tns:GetInboxRulesSoapIn Message

The **GetInboxRulesSoapIn** WSDL message specifies a request to get Inbox rules (2).

```

<wsdl:message name="GetInboxRulesSoapIn">
  <wsdl:part name="request" element="tns:GetInboxRules" />
  <wsdl:part name="Impersonation" element="t:ExchangeImpersonation" />
  <wsdl:part name="MailboxCulture" element="t:MailboxCulture" />
  <wsdl:part name="RequestVersion" element="t:RequestServerVersion" />
  <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext" />
</wsdl:message>

```

The **GetInboxRulesSoapIn** WSDL message is the input message for the **SOAP action** <http://schemas.microsoft.com/exchange/services/2006/messages/GetInboxRules>.

The five parts of the **GetInboxRulesSoapIn** message are described in the following table.

Part name	Element/type	Description
request	tns:GetInboxRules (section 3.1.4.1.2.1)	Specifies the SOAP body of the request to get the Inbox rules (2).
Impersonation	t:ExchangeImpersonation [MS-OXWSCDATA] section 2.2.4.32)	Specifies a SOAP header that identifies the user whom the client application is impersonating.

Part name	Element/type	Description
MailboxCulture	t:MailboxCulture ([MS-OXWSCDATA] section 2.2.4.43)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by RFC3066 .
RequestVersion	t:RequestServerVersion ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the GetInboxRules operation request.
TimeZoneContext	t:TimeZoneContext ([MS-OXWSGTZ] section 2.2.3.4)	Specifies a time zone definition and enables the association of SOAP attributes with the definition.

3.1.4.1.1.2 tns:GetInboxRulesSoapOut Message

The **GetInboxRulesSoapOut** WSDL message specifies the server response to a request to get Inbox rules (2).

```
<wsdl:message name="GetInboxRulesSoapOut">
  <wsdl:part name="GetInboxRulesResult" element="tns:GetInboxRulesResponse" />
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo" />
</wsdl:message>
```

The **GetInboxRulesSoapOut** WSDL message is the output message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/GetInboxRules>.

The two parts of the **GetInboxRulesSoapOut** WSDL message are described in the following table.

Part name	Element/type	Description
GetInboxRulesResult	tns:GetInboxRulesResponse (section 3.1.4.1.2.2)	Specifies the SOAP body of the response to a GetInboxRules operation request.
ServerVersion	t:ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.1.2 Elements

The following table summarizes the XML schema element definitions are specific to this operation.

Element name	Description
GetInboxRules	Specifies a request to get the Inbox rules (2) in a mailbox in the server message store.
GetInboxRulesResponse	Specifies the response body content from a request to get Inbox rules (2).

3.1.4.1.2.1 GetInboxRules Element

The **GetInboxRules** element specifies a request message for a **GetInboxRules** operation.

```
<xs:element name="GetInboxRules"
  type="m:GetInboxRulesRequestType"
/>
```

3.1.4.1.2.2 GetInboxRulesResponse Element

The **GetInboxRulesResponse** element specifies a response message for a **GetInboxRules** operation.

```
<xs:element name="GetInboxRulesResponse"
  type="m:GetInboxRulesResponseType"
/>
```

3.1.4.1.3 Complex Types

The following table summarizes the XML schema complex type definitions that are specific to this operation.

Complex type name	Description
GetInboxRulesRequestType	Specifies a request message for the GetInboxRules operation.
GetInboxRulesResponseType	Specifies a response message for the GetInboxRules operation.
ArrayOfRulesType	Specifies an array of RuleType complex type objects.

3.1.4.1.3.1 m:GetInboxRulesRequestType Complex Type

The **GetInboxRulesRequestType** complex type specifies a request message to get the Inbox rules (2) in a server database. The **GetInboxRulesRequestType** complex type extends the **BaseRequestType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.17.

```
<xs:complexType name="GetInboxRulesRequestType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="MailboxSmtAddress"
          type="xs:string"
          minOccurs="0"
          maxOccurs="1"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child element of the **GetInboxRulesRequestType** complex type.

Element name	Type	Description
MailboxSmtAddress	xs:string (XMLSCHEMA21)	Represents the SMTP address of the user whose Inbox rules (2) are to be retrieved.

3.1.4.1.3.2 m:GetInboxRulesResponseType Complex Type

The **GetInboxRulesResponseType** complex type specifies the response message that is returned by the **GetInboxRules** operation. The **GetInboxRulesResponseType** complex type extends the **ResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.65.

```
<xs:complexType name="GetInboxRulesResponseType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="OutlookRuleBlobExists"
          type="xs:boolean"
          minOccurs="0"
          maxOccurs="1"
        />
        <xs:element name="InboxRules"
          type="t:ArrayOfRulesType"
          minOccurs="0"
          maxOccurs="1"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **GetInboxRulesResponseType** complex type.

Element name	Type	Description
OutlookRuleBlobExists	xs:boolean (XMLSCHEMA21)	Specifies whether a binary large object (BLOB) that contains all rules (2) exists in the user's mailbox. A text value of "true" indicates that the BLOB exists. This element MUST be present.
InboxRules	t:ArrayOfRulesType (section 3.1.4.1.3.3)	Represents the response message and includes an array of the rules (2) in the user's mailbox.

3.1.4.1.3.3 t:ArrayOfRulesType Complex Type

The **ArrayOfRulesType** complex type specifies an array of **RuleType** complex type, as specified in section [2.2.4.5](#).

```
<xs:complexType name="ArrayOfRulesType">
  <xs:sequence>
    <xs:element name="Rule"
      type="t:RuleType"
      minOccurs="0"
      maxOccurs="unbounded"
    />
  </xs:sequence>
</xs:complexType>
```

```

    />
  </xs:sequence>
</xs:complexType>

```

The following table lists the child element of the **ArrayOfRulesType** complex type.

Element name	Type	Description
Rule	t:RuleType (section 2.2.4.5)	The rule (2) in the user's mailbox.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 UpdateInboxRules

The **UpdateInboxRules** operation updates the authenticated user's Inbox rules (2) by applying the specified operations. This operation is used to create an Inbox rule (2), to set an Inbox rule (2), or to delete an Inbox rule (2).

The following is the WSDL port type specification of the operation.

```

<wsdl:operation name="UpdateInboxRules">
  <wsdl:input message="tns:UpdateInboxRulesSoapIn" />
  <wsdl:output message="tns:UpdateInboxRulesSoapOut" />
</wsdl:operation>

```

The following is the WSDL binding specification of the operation.

```

<wsdl:operation name="UpdateInboxRules">
  <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/UpdateInboxRules" />
  <wsdl:input>
    <soap:header message="tns:UpdateInboxRulesSoapIn" part="Impersonation" use="literal"
/>
    <soap:header message="tns:UpdateInboxRulesSoapIn" part="MailboxCulture" use="literal"
/>
    <soap:header message="tns:UpdateInboxRulesSoapIn" part="RequestVersion" use="literal"
/>
    <soap:header message="tns:UpdateInboxRulesSoapIn" part="TimeZoneContext"
use="literal" />
  </wsdl:input>
</wsdl:operation>

```

```

    <soap:body parts="request" use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="UpdateInboxRulesResult" use="literal" />
    <soap:header message="tns:UpdateInboxRulesSoapOut" part="ServerVersion" use="literal"
  />
  </wsdl:output>
</wsdl:operation>

```

The protocol client sends an **UpdateInboxRulesSoapIn** request WSDL message and the protocol server responds with an **UpdateInboxRulesSoapOut** response WSDL message.

A successful **UpdateInboxRules** operation request returns an **UpdateInboxRulesResponse** element with the **ResponseClass** attribute of the **UpdateInboxRulesResponseType** element set to "Success". The **ResponseCode** element of the **UpdateInboxRulesResponse** element is set to "NoError".

3.1.4.2.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message name	Description
UpdateInboxRulesSoapIn	Specifies the SOAP message that updates the Inbox rules (2).
UpdateInboxRulesSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.2.1.1 tns:UpdateInboxRulesSoapIn Message

The **UpdateInboxRulesSoapIn** WSDL message specifies a request to update Inbox rules (2). The update can be to create, set, or delete the Inbox rule (2).

```

<wsdl:message name="UpdateInboxRulesSoapIn">
  <wsdl:part name="request" element="tns:UpdateInboxRules" />
  <wsdl:part name="Impersonation" element="t:ExchangeImpersonation" />
  <wsdl:part name="MailboxCulture" element="t:MailboxCulture" />
  <wsdl:part name="RequestVersion" element="t:RequestServerVersion" />
  <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext" />
</wsdl:message>

```

The **UpdateInboxRulesSoapIn** WSDL message is the input message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/UpdateInboxRules>.

The five parts of the **UpdateInboxRulesSoapIn** message are described in the following table.

Part name	Element/type	Description
request	tns:UpdateInboxRules (section 3.1.4.2.2.1)	Specifies the SOAP body of the request to update Inbox rules (2).
Impersonation	t:ExchangeImpersonation ([MS-OXWSCDATA] section 2.2.4.32)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
MailboxCulture	t:MailboxCulture ([MS-OXWSCDATA])	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are

Part name	Element/type	Description
	section 2.2.4.43)	defined by RFC3066 .
RequestVersion	t:RequestServerVersion ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the UpdateInboxRules operation request.
TimeZoneContext	t:TimeZoneContext ([MS-OXWSGTZ] section 2.2.3.4)	Specifies a time zone definition and enables the association of SOAP attributes with the definition.

3.1.4.2.1.2 tns:UpdateInboxRulesSoapOut Message

The **UpdateInboxRulesSoapOut** WSDL message specifies the server response to the **UpdateInboxRules** operation request to update the Inbox rules (2).

```
<wsdl:message name="UpdateInboxRulesSoapOut">
  <wsdl:part name="UpdateInboxRulesResult" element="tns:UpdateInboxRulesResponse" />
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo" />
</wsdl:message>
```

The **UpdateInboxRulesSoapOut** WSDL message is the output message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/UpdateInboxRules>.

The two parts of the **UpdateInboxRulesSoapOut** WSDL message are described in the following table.

Part name	Element/type	Description
UpdateInboxRulesResult	tns:UpdateInboxRulesResponse (section 3.1.4.2.2.2)	Specifies the response to an UpdateInboxRules operation request.
ServerVersion	t:ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.12)	Specifies the server version for the response.

3.1.4.2.2 Elements

The following table summarizes the XML schema element definitions are specific to this operation.

Element name	Description
UpdateInboxRules	Specifies a request to update the Inbox rules (2) in a mailbox in the server message store.
UpdateInboxRulesResponse	Specifies the response body content from a request to update the Inbox rules (2).

3.1.4.2.2.1 UpdateInboxRules Element

The **UpdateInboxRules** element specifies the request message for an **UpdateInboxRules** operation.


```
<xs:element name="UpdateInboxRules"
  type="m:UpdateInboxRulesRequestType"
 />
```

3.1.4.2.2 UpdateInboxRulesResponse Element

The **UpdateInboxRulesResponse** element specifies the response message for an **UpdateInboxRules** operation.

```
<xs:element name="UpdateInboxRulesResponse"
  type="m:UpdateInboxRulesResponseType"
 />
```

3.1.4.2.3 Complex Types

The following table summarizes the XML schema complex type definitions that are specific to this operation.

Complex type name	Description
UpdateInboxRulesRequestType	Specifies the request message for the UpdateInboxRules operation.
UpdateInboxRulesResponseType	Specifies the response message for the UpdateInboxRules operation.
ArrayOfRuleOperationsType	Specifies an array of rule (2) operations that can be performed on a rule (2).
RuleOperationType	Specifies the base type for rule (2) operations and represents the rule (2) operation to be performed.
CreateRuleOperationType	Specifies an operation to create a new rule (2).
SetRuleOperationType	Specifies an operation to update an existing rule (2).
DeleteRuleOperationType	Specifies an operation to delete an existing rule (2).
ArrayOfRuleOperationErrorsType	Specifies an array of rule (2) validation errors on each rule (2) field that has an error.
RuleOperationErrorType	Specifies a rule (2) operation error.
ArrayOfRuleValidationErrorsType	Specifies an array of rule (2) validation errors.
RuleValidationErrorType	Specifies a single validation error on a particular rule (2) property value, predicate property value, or action property value.

3.1.4.2.3.1 m:UpdateInboxRulesRequestType Complex Type

The **UpdateInboxRulesRequestType** complex type specifies a request message to update the Inbox rules (2) in a server database. The **UpdateInboxRulesRequestType** complex type extends the **BaseRequestType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.17.

```
<xs:complexType name="UpdateInboxRulesRequestType">
  <xs:complexContent>
```

```

<xs:extension
  base="m:BaseRequestType"
>
  <xs:sequence>
    <xs:element name="MailboxSmtpAddress"
      type="xs:string"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="RemoveOutlookRuleBlob"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
    />
    <xs:element name="Operations"
      type="t:ArrayOfRuleOperationsType"
      minOccurs="1"
      maxOccurs="1"
    />
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

The following table lists the child elements of the **UpdateInboxRulesRequestType** complex type.

Element name	Type	Description
MailboxSmtpAddress	xs:string ([XMLSCHEMA2])	Represents the SMTP address of the user whose Inbox rules (2) are to be updated.
RemoveOutlookRuleBlob	xs:boolean ([XMLSCHEMA2])	Indicates whether to remove the BLOB that contains all rules (2).
Operations	t:ArrayOfRuleOperationsType (section 3.1.4.2.3.3)	The operations that are to be applied to the user's Inbox rules (2).

3.1.4.2.3.2 m:UpdateInboxRulesResponseType Complex Type

The **UpdateInboxRulesResponseType** complex type specifies a response message that is returned by the **UpdateInboxRules** operation. The **UpdateInboxRulesResponseType** complex type extends the **ResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.65.

```

<xs:complexType name="UpdateInboxRulesResponseType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="RuleOperationErrors"
          type="t:ArrayOfRuleOperationErrorsType"
          minOccurs="0"
          maxOccurs="1"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The following table lists the child element of the **UpdateInboxRulesResponseType** complex type.

Element name	Type	Description
RuleOperationErrors	t:ArrayOfRuleOperationErrorsType (section 3.1.4.2.3.8)	Specifies a RuleOperationErrorType complex type (section 3.1.4.2.3.9) collection, which contains the validation errors on each rule (2) filed that has an error.

3.1.4.2.3.3 t:ArrayOfRuleOperationsType Complex Type

The **ArrayOfRuleOperationsType** complex type specifies an array of rule (2) operations that can be performed on a rule (2).

```
<xs:complexType name="ArrayOfRuleOperationsType">
  <xs:choice
    maxOccurs="unbounded"
    minOccurs="1"
  >
    <xs:element name="CreateRuleOperation"
      type="t:CreateRuleOperationType"
    />
    <xs:element name="SetRuleOperation"
      type="t:SetRuleOperationType"
    />
    <xs:element name="DeleteRuleOperation"
      type="t>DeleteRuleOperationType"
    />
  </xs:choice>
</xs:complexType>
```

The following table lists the child elements of the **ArrayOfRuleOperationsType** complex type.

Element name	Type	Description
CreateRuleOperation	t:CreateRuleOperationType (section 3.1.4.2.3.5)	Specifies an operation to create an Inbox rule (2).
SetRuleOperation	t:SetRuleOperationType (section 3.1.4.2.3.6)	Specifies an operation to update an Inbox rule (2).
DeleteRuleOperation	t>DeleteRuleOperationType (section 3.1.4.2.3.7)	Specifies an operation to delete an Inbox rule (2).

3.1.4.2.3.4 t:RuleOperationType Complex Type

The **RuleOperationType** complex type is the base type for rule (2) operations and represents the rule (2) operation to be performed.

```
<xs:complexType name="RuleOperationType"
  abstract="true"
/>
```

The **CreateRuleOperationType** complex type, as specified in section [3.1.4.2.3.5](#), **SetRuleOperationType** complex type, as specified in section [3.1.4.2.3.6](#), and **DeleteRuleOperationType** complex type, as specified in section [3.1.4.2.3.7](#), extend this base type.

3.1.4.2.3.5 t:CreateRuleOperationType Complex Type

The **CreateRuleOperationType** complex type represents the operation to create a new Inbox rule (2). The **CreateRuleOperationType** complex type extends the **RuleOperationType** complex type, as specified in section [3.1.4.2.3.4](#).

```
<xs:complexType name="CreateRuleOperationType">
  <xs:complexContent>
    <xs:extension
      base="t:RuleOperationType"
    >
      <xs:sequence>
        <xs:element name="Rule"
          type="t:RuleType"
          maxOccurs="1"
          minOccurs="1"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child element of the **CreateRuleOperationType** complex type.

Element name	Type	Description
Rule	t:RuleType (section 2.2.4.5)	Represents the rule (2) to create.

The **UpdateInboxRules** operation can be used with the **CreateRuleOperationType** complex type to create a rule (2). When creating a rule (2), the **RuleType** complex type, as specified in section 2.2.4.5, MUST NOT have a **RuleId** element. The **RuleId** element of the **RuleType** complex type is generated when the rule (2) is successfully created in the server message store.

3.1.4.2.3.6 t:SetRuleOperationType Complex Type

The **SetRuleOperationType** complex type represents an operation to update an existing Inbox rule (2). The **SetRuleOperationType** complex type extends the **RuleOperationType** complex type, as specified in section [3.1.4.2.3.4](#).

```
<xs:complexType name="SetRuleOperationType">
  <xs:complexContent>
    <xs:extension
      base="t:RuleOperationType"
    >
      <xs:sequence>
        <xs:element name="Rule"
          type="t:RuleType"
          maxOccurs="1"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

        minOccurs="1"
      />
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>

```

The following table lists the child element of the **SetRuleOperationType** complex type.

Element name	Type	Description
Rule	t:RuleType (section 2.2.4.5)	Represents the rule (2) to be updated.

The **UpdateInboxRules** operation can be used with the **SetRuleOperationType** complex type to modify a rule (2). When modifying a rule (2), the **RuleType** complex type **MUST** have a **RuleId** element.

3.1.4.2.3.7 t:DeleteRuleOperationType Complex Type

The **DeleteRuleOperationType** complex type represents an operation to delete an existing Inbox rule (2). The **DeleteRuleOperationType** complex type extends the **RuleOperationType** complex type, as specified in section [3.1.4.2.3.4](#).

```

<xs:complexType name="DeleteRuleOperationType">
  <xs:complexContent>
    <xs:extension
      base="t:RuleOperationType"
    >
      <xs:sequence>
        <xs:element name="RuleId"
          type="xs:string"
          minOccurs="1"
          maxOccurs="1"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The following table lists the child element of the **DeleteRuleOperationType** complex type.

Element name	Type	Description
RuleId	xs:string ([XMLSCHEMA2])	Specifies the rule (2) identifier of the Inbox rule (2) to be deleted.

3.1.4.2.3.8 t:ArrayOfRuleOperationErrorsType Complex Type

The **ArrayOfRuleOperationErrorsType** complex type specifies an array of rule (2) validation errors on each rule (2) field that has an error.

```

<xs:complexType name="ArrayOfRuleOperationErrorsType">

```

```

<xs:sequence>
  <xs:element name="RuleOperationError"
    type="t:RuleOperationErrorType"
    maxOccurs="unbounded"
    minOccurs="1"
  />
</xs:sequence>
</xs:complexType>

```

The following table lists the child element of the **ArrayOfRuleOperationErrorsType** complex type.

Element name	Type	Description
RuleOperationError	t:RuleOperationErrorType (section 3.1.4.2.3.9)	Specifies a rule (2) operation error.

3.1.4.2.3.9 t:RuleOperationErrorType Complex Type

The **RuleOperationErrorType** complex type specifies a rule (2) operation error.

```

<xs:complexType name="RuleOperationErrorType">
  <xs:sequence>
    <xs:element name="OperationIndex"
      type="xs:int"
      maxOccurs="1"
      minOccurs="1"
    />
    <xs:element name="ValidationErrors"
      type="t:ArrayOfRuleValidationErrorsType"
      maxOccurs="1"
      minOccurs="1"
    />
  </xs:sequence>
</xs:complexType>

```

The following table lists the child elements of the **RuleOperationErrorType** complex type.

Element name	Type	Description
OperationIndex	xs:int ([XMLSCHEMA2])	Specifies the index of the operation in the request that caused the rule (2) operation error.
ValidationErrors	t:ArrayOfRuleValidationErrorsType (section 3.1.4.2.3.10)	Specifies an array of rule (2) validation errors on each rule (2) field that has an error.

3.1.4.2.3.10 t:ArrayOfRuleValidationErrorsType Complex Type

The **ArrayOfRuleValidationErrorsType** complex type specifies an array of rule (2) validation errors.

```

<xs:complexType name="ArrayOfRuleValidationErrorsType">
  <xs:sequence>

```

```

<xs:element name="Error"
  type="t:RuleValidationErrorType"
  maxOccurs="unbounded"
  minOccurs="1"
/>
</xs:sequence>
</xs:complexType>

```

The following table lists the child element of the **ArrayOfRuleValidationErrorsType** complex type.

Element name	Type	Description
Error	t:RuleValidationErrorType (section 3.1.4.2.3.11)	Specifies a single validation error on a particular rule (2) property value, predicate property value, or action property value.

3.1.4.2.3.11 t:RuleValidationErrorType Complex Type

The **RuleValidationErrorType** complex type specifies a single validation error on a particular rule (2) property value, predicate property value, or action property value.

```

<xs:complexType name="RuleValidationErrorType">
  <xs:sequence>
    <xs:element name="FieldURI"
      type="t:RuleFieldURIType"
      minOccurs="1"
      maxOccurs="1"
    />
    <xs:element name="ErrorCode"
      type="t:RuleValidationErrorCodeType"
      minOccurs="1"
      maxOccurs="1"
    />
    <xs:element name="ErrorMessage"
      type="xs:string"
      minOccurs="1"
      maxOccurs="1"
    />
    <xs:element name="FieldValue"
      type="xs:string"
      minOccurs="1"
      maxOccurs="1"
    />
  </xs:sequence>
</xs:complexType>

```

The following table lists the child elements of the **RuleValidationErrorType** complex type.

Element name	Type	Description
FieldURI	t:RuleFieldURIType (section 3.1.4.2.4.1)	Specifies the URI to the rule (2) field that caused the validation error.
ErrorCode	t:RuleValidationErrorCodeType (section 3.1.4.2.4.2)	Specifies the error code that was triggered on

Element name	Type	Description
	3.1.4.2.4.2)	the field.
ErrorMessage	xs:string ([XMLSCHEMA2])	Specifies the reason for the validation error.
FieldValue	xs:string	Specifies the value of the field that caused the validation error. This element is optional.

3.1.4.2.4 Simple Types

The following table summarizes the XML schema simple types that are specific to this operation.

Simple type name	Description
RuleFieldURIType	Defines the rule (2) fields that could cause a validation error.
RuleValidationErrorCodeType	Defines what failed validation for each rule (2) predicate or action.

3.1.4.2.4.1 t:RuleFieldURIType Simple Type

The **RuleFieldURIType** simple type defines all possible rule (2) fields that could trigger a validation error.

```
<xs:simpleType name="RuleFieldURIType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="RuleId"
    />
    <xs:enumeration
      value="DisplayName"
    />
    <xs:enumeration
      value="Priority"
    />
    <xs:enumeration
      value="IsNotSupported"
    />
    <xs:enumeration
      value="Actions"
    />
    <xs:enumeration
      value="Condition:Categories"
    />
    <xs:enumeration
      value="Condition:ContainsBodyStrings"
    />
    <xs:enumeration
      value="Condition:ContainsHeaderStrings"
    />
    <xs:enumeration
      value="Condition:ContainsRecipientStrings"
    />
    <xs:enumeration
      value="Condition:ContainsSenderStrings"
    />
  </xs:restriction>
</xs:simpleType>
```



```
<xs:enumeration
  value="Condition:ContainsSubjectOrBodyStrings"
/>
<xs:enumeration
  value="Condition:ContainsSubjectStrings"
/>
<xs:enumeration
  value="Condition:FlaggedForAction"
/>
<xs:enumeration
  value="Condition:FromAddresses"
/>
<xs:enumeration
  value="Condition:FromConnectedAccounts"
/>
<xs:enumeration
  value="Condition:HasAttachments"
/>
<xs:enumeration
  value="Condition:Importance"
/>
<xs:enumeration
  value="Condition:IsApprovalRequest"
/>
<xs:enumeration
  value="Condition:IsAutomaticForward"
/>
<xs:enumeration
  value="Condition:IsAutomaticReply"
/>
<xs:enumeration
  value="Condition:IsEncrypted"
/>
<xs:enumeration
  value="Condition:IsMeetingRequest"
/>
<xs:enumeration
  value="Condition:IsMeetingResponse"
/>
<xs:enumeration
  value="Condition:IsNDR"
/>
<xs:enumeration
  value="Condition:IsPermissionControlled"
/>
<xs:enumeration
  value="Condition:IsReadReceipt"
/>
<xs:enumeration
  value="Condition:IsSigned"
/>
<xs:enumeration
  value="Condition:IsVoicemail"
/>
<xs:enumeration
  value="Condition:ItemClasses"
/>
<xs:enumeration
  value="Condition:MessageClassifications"
/>
<xs:enumeration
  value="Condition:NotSentToMe"
/>
<xs:enumeration
  value="Condition:SentCcMe"
/>
<xs:enumeration
  value="Condition:SentOnlyToMe"
```

```

/>
<xs:enumeration
  value="Condition:SentToAddresses"
/>
<xs:enumeration
  value="Condition:SentToMe"
/>
<xs:enumeration
  value="Condition:SentToOrCcMe"
/>
<xs:enumeration
  value="Condition:Sensitivity"
/>
<xs:enumeration
  value="Condition:WithinDateRange"
/>
<xs:enumeration
  value="Condition:WithinSizeRange"
/>
<xs:enumeration
  value="Exception:Categories"
/>
<xs:enumeration
  value="Exception:ContainsBodyStrings"
/>
<xs:enumeration
  value="Exception:ContainsHeaderStrings"
/>
<xs:enumeration
  value="Exception:ContainsRecipientStrings"
/>
<xs:enumeration
  value="Exception:ContainsSenderStrings"
/>
<xs:enumeration
  value="Exception:ContainsSubjectOrBodyStrings"
/>
<xs:enumeration
  value="Exception:ContainsSubjectStrings"
/>
<xs:enumeration
  value="Exception:FlaggedForAction"
/>
<xs:enumeration
  value="Exception:FromAddresses"
/>
<xs:enumeration
  value="Exception:FromConnectedAccounts"
/>
<xs:enumeration
  value="Exception:HasAttachments"
/>
<xs:enumeration
  value="Exception:Importance"
/>
<xs:enumeration
  value="Exception:IsApprovalRequest"
/>
<xs:enumeration
  value="Exception:IsAutomaticForward"
/>
<xs:enumeration
  value="Exception:IsAutomaticReply"
/>
<xs:enumeration
  value="Exception:IsEncrypted"
/>
<xs:enumeration

```

```

    value="Exception:IsMeetingRequest"
  />
<xs:enumeration
  value="Exception:IsMeetingResponse"
 />
<xs:enumeration
  value="Exception:IsNDR"
 />
<xs:enumeration
  value="Exception:IsPermissionControlled"
 />
<xs:enumeration
  value="Exception:IsReadReceipt"
 />
<xs:enumeration
  value="Exception:IsSigned"
 />
<xs:enumeration
  value="Exception:IsVoicemail"
 />
<xs:enumeration
  value="Exception:ItemClasses"
 />
<xs:enumeration
  value="Exception:MessageClassifications"
 />
<xs:enumeration
  value="Exception:NotSentToMe"
 />
<xs:enumeration
  value="Exception:SentCcMe"
 />
<xs:enumeration
  value="Exception:SentOnlyToMe"
 />
<xs:enumeration
  value="Exception:SentToAddresses"
 />
<xs:enumeration
  value="Exception:SentToMe"
 />
<xs:enumeration
  value="Exception:SentToOrCcMe"
 />
<xs:enumeration
  value="Exception:Sensitivity"
 />
<xs:enumeration
  value="Exception:WithinDateRange"
 />
<xs:enumeration
  value="Exception:WithinSizeRange"
 />
<xs:enumeration
  value="Action:AssignCategories"
 />
<xs:enumeration
  value="Action:CopyToFolder"
 />
<xs:enumeration
  value="Action>Delete"
 />
<xs:enumeration
  value="Action:ForwardAsAttachmentToRecipients"
 />
<xs:enumeration
  value="Action:ForwardToRecipients"
 />

```

```

<xs:enumeration
  value="Action:MarkImportance"
/>
<xs:enumeration
  value="Action:MarkAsRead"
/>
<xs:enumeration
  value="Action:MoveToFolder"
/>
<xs:enumeration
  value="Action:PermanentDelete"
/>
<xs:enumeration
  value="Action:RedirectToRecipients"
/>
<xs:enumeration
  value="Action:SendSMSAlertToRecipients"
/>
<xs:enumeration
  value="Action:ServerReplyWithMessage"
/>
<xs:enumeration
  value="Action:StopProcessingRules"
/>
<xs:enumeration
  value="IsEnabled"
/>
<xs:enumeration
  value="IsInError"
/>
<xs:enumeration
  value="Conditions"
/>
<xs:enumeration
  value="Exceptions"
/>
</xs:restriction>
</xs:simpleType>

```

The following table lists the values that are defined by the **RuleFieldURIType** simple type.

Value name	Meaning
RuleId	The rule (2) ID.
DisplayName	The rule (2) display name.
Priority	The rule (2) priority.
IsNotSupported	The rule (2) is not supported.
Actions	The rule (2) actions.
Condition:Categories	The categories condition.
Condition:ContainsBodyStrings	The body contains strings condition.
Condition:ContainsHeaderStrings	The header contains strings condition.
Condition:ContainsRecipientStrings	The recipient contains strings condition.
Condition:ContainsSenderStrings	The sender contains strings condition.

Value name	Meaning
Condition:ContainsSubjectOrBodyStrings	The subject or body contains strings condition.
Condition:ContainsSubjectStrings	The subject contains strings condition.
Condition:FlaggedForAction	The flagged for action condition.
Condition:FromAddresses	The from addresses condition.
Condition:FromConnectedAccounts	The from connected accounts condition.
Condition:HasAttachments	The has attachments condition.
Condition:Importance	The importance condition property of a rule's (2) set of conditions.
Condition:IsApprovalRequest	The approval request condition exists.
Condition:IsAutomaticForward	The automatic forward condition exists.
Condition:IsAutomaticReply	The automatic reply condition exists.
Condition:IsEncrypted	The encrypted condition exists.
Condition:IsMeetingRequest	The meeting request condition exists.
Condition:IsMeetingResponse	The meeting response condition exists.
Condition:IsNDR	The non-delivery report (NDR) condition exists.
Condition:IsPermissionControlled	The permission-controlled condition exists.
Condition:IsReadReceipt	The read condition exists.
Condition:IsSigned	The signed condition exists.
Condition:IsVoicemail	The voice mail condition exists.
Condition:ItemClasses	The item class condition.
Condition:MessageClassifications	The message classifications condition.
Condition:NotSentToMe	The not sent to me condition.
Condition:SentCcMe	The sent Cc me condition.
Condition:SentOnlyToMe	The sent only to me condition.
Condition:SentToAddresses	The sent to addresses condition.
Condition:SentToMe	The sent to me condition.
Condition:SentToOrCcMe	The sent to or Cc me condition.
Condition:Sensitivity	The sensitivity condition.
Condition:WithinDateRange	The within date range condition.
Condition:WithinSizeRange	The within size range condition.
Exception:Categories	The categories exception.
Exception:ContainsBodyStrings	The body contains strings exception.

Value name	Meaning
Exception:ContainsHeaderStrings	The header contains strings exception.
Exception:ContainsRecipientStrings	The recipient contains strings exception.
Exception:ContainsSenderStrings	The sender contains strings exception.
Exception:ContainsSubjectOrBodyStrings	The subject or body contains strings exception.
Exception:ContainsSubjectStrings	The subject contains strings exception.
Exception:FlaggedForAction	The flagged for action exception.
Exception:FromAddresses	The from addresses exception.
Exception:FromConnectedAccounts	The from connected accounts exception.
Exception:HasAttachments	The has attachments exception.
Exception:Importance	The importance exception.
Exception:IsApprovalRequest	The approval request exception exists.
Exception:IsAutomaticForward	The automatic forward exception exists
Exception:IsAutomaticReply	The automatic reply exception exists.
Exception:IsEncrypted	The encrypted exception exists.
Exception:IsMeetingRequest	The meeting request exception exists.
Exception:IsMeetingResponse	The meeting response exception exists.
Exception:IsNDR	The NDR exception exists.
Exception:IsPermissionControlled	The permission controlled exception exists.
Exception:IsReadReceipt	The read exception exists.
Exception:IsSigned	The signed exception exists.
Exception:IsVoicemail	The voice mail exception exists.
Exception:ItemClasses	The item class exception.
Exception:MessageClassifications	The message classifications exception.
Exception:NotSentToMe	The not sent to me exception.
Exception:SentCcMe	The sent to me exception.
Exception:SentOnlyToMe	The sent only to me exception.
Exception:SentToAddresses	The sent to addresses exception.
Exception:SentToMe	The sent to me exception.
Exception:SentToOrCcMe	The sent to or Cc me exception.
Exception:Sensitivity	The sensitivity exception.
Exception:WithinDateRange	The within date range exception.

Value name	Meaning
Exception:WithinSizeRange	The within size range exception.
Action:AssignCategories	The assign categories action.
Action:CopyToFolder	The copy to folder action.
Action>Delete	The delete action.
Action:ForwardAsAttachmentToRecipients	The forward as attachment to recipients action.
Action:ForwardToRecipients	The forward to recipients action.
Action:MarkImportance	The mark importance action.
Action:MarkAsRead	The mark as read action.
Action:MoveToFolder	The move to folder action.
Action:PermanentDelete	The permanent delete action.
Action:RedirectToRecipients	The redirect to recipients action.
Action:SendSMSAlertToRecipients	The send SMS alert to recipients action.
Action:ServerReplyWithMessage	The server reply with message action.
Action:StopProcessingRules	The stop processing rules action.
IsEnabled	Indicates whether the rule (2) is enabled.
IsInError	Indicates whether the rule (2) is in error.
Conditions	Contains all conditions of the rule (2).
Exceptions	Contains all the exceptions for the rule (2).

3.1.4.2.4.2 t:RuleValidationErrorCodeType Simple Type

The **RuleValidationErrorCodeType** simple type defines what failed validation for each rule (2) predicate or action.

```

<xs:simpleType name="RuleValidationErrorCodeType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="ADOperationFailure"
    />
    <xs:enumeration
      value="ConnectedAccountNotFound"
    />
    <xs:enumeration
      value="CreateWithRuleId"
    />
    <xs:enumeration
      value="EmptyValueFound"
    />
    <xs:enumeration
      value="DuplicatedPriority"
    />
  </xs:restriction>
</xs:simpleType>

```

```

/>
<xs:enumeration
  value="DuplicatedOperationOnTheSameRule"
/>
<xs:enumeration
  value="FolderDoesNotExist"
/>
<xs:enumeration
  value="InvalidAddress"
/>
<xs:enumeration
  value="InvalidDateRange"
/>
<xs:enumeration
  value="InvalidFolderId"
/>
<xs:enumeration
  value="InvalidSizeRange"
/>
<xs:enumeration
  value="InvalidValue"
/>
<xs:enumeration
  value="MessageClassificationNotFound"
/>
<xs:enumeration
  value="MissingAction"
/>
<xs:enumeration
  value="MissingParameter"
/>
<xs:enumeration
  value="MissingRangeValue"
/>
<xs:enumeration
  value="NotSettable"
/>
<xs:enumeration
  value="RecipientDoesNotExist"
/>
<xs:enumeration
  value="RuleNotFound"
/>
<xs:enumeration
  value="SizeLessThanZero"
/>
<xs:enumeration
  value="StringValueTooBig"
/>
<xs:enumeration
  value="UnsupportedAddress"
/>
<xs:enumeration
  value="UnexpectedError"
/>
<xs:enumeration
  value="UnsupportedRule"
/>
</xs:restriction>
</xs:simpleType>

```

The following table lists the values that are defined by the **RuleValidationErrorCodeType** simple type.

Value name	Meaning
ADOperationFailure	The directory service operation failed.
ConnectedAccountNotFound	The specified email account was not found.
CreateWithRuleId	An attempt was made to create a rule (2) with an ID. The IDs of new rules (2) are generated by the server and are not provided by the client.
EmptyValueFound	The Value field is empty. An empty Value field is not allowed.
DuplicatedPriority	There are multiple rules (2) in the same request with the same priority. A priority can have only one rule (2).
DuplicatedOperationOnTheSameRule	There are multiple operations against the same rule (2). Only one operation per rule (2) is allowed.
FolderDoesNotExist	The folder does not exist in the user's mailbox.
InvalidAddress	The email address is invalid.
InvalidDateRange	The specified date range is invalid.
InvalidFolderId	The specified folder ID is invalid.
InvalidSizeRange	The size range is invalid.
InvalidValue	The specified value is invalid.
MessageClassificationNotFound	The message classification was not found.
MissingAction	No action was specified. At least one rule (2) action has to be specified for validation.
MissingParameter	The required parameter is missing.
MissingRangeValue	The range value is missing.
NotSettable	The property cannot be modified.
RecipientDoesNotExist	The recipient does not exist.
RuleNotFound	The rule (2) was not found. No rules (2) were found that match the criteria.
SizeLessThanZero	The specified size is less than zero.
StringValueTooBig	The string value is too big.
UnsupportedAddress	The address is not supported. For example, Inbox rules (2) do not support MAIPDL address types.
UnexpectedError	An unexpected error occurred.
UnsupportedRule	The rule (2) is unsupported.

When a rule (2) is generated with the same priority as an existing rule (2), the server can increment the priority of the existing rule (2) to eliminate the error that is indicated by the "DuplicatedPriority" value.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

Preliminary

4 Protocol Examples

The following examples show the request and response XML for the Inbox Rules Web Service Protocol operations.

4.1 CreateInboxRule

The following is an example of an **UpdateInboxRules** operation that creates a rule (2).

The client constructs the request **XML** and sends it to the server. The **CreateRuleOperation** complex type in the request contains the rule (2) to be created. This example creates a rule (2) that will move an email message to the **Junk Email folder** if the email subject contains a string that equals "Interesting".

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010_SP1" />
  </soap:Header>
  <soap:Body>
    <m:UpdateInboxRules>
      <m:RemoveOutlookRuleBlob>true</m:RemoveOutlookRuleBlob>
      <m:Operations>
        <t:CreateRuleOperation>
          <t:Rule>
            <t:DisplayName>MoveInterestingToJunk</t:DisplayName>
            <t:Priority>1</t:Priority>
            <t:IsEnabled>true</t:IsEnabled>
            <t:IsInError>>false</t:IsInError>
            <t:Conditions>
              <t:ContainsSubjectStrings>
                <t:String>Interesting</t:String>
              </t:ContainsSubjectStrings>
            </t:Conditions>
            <t:Exceptions />
            <t:Actions>
              <t:MoveToFolder>
                <t:DistinguishedFolderId Id="junkemail" />
              </t:MoveToFolder>
            </t:Actions>
          </t:Rule>
        </t:CreateRuleOperation>
      </m:Operations>
    </m:UpdateInboxRules>
  </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
      MinorVersion="1"
      MajorBuildNumber="139"
      MinorBuildNumber="0"
      Version="Exchange2010_SP1"
```

```

    xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
</s:Header>
<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <UpdateInboxRulesResponse ResponseClass="Success"
    xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
    <ResponseCode>NoError</ResponseCode>
  </UpdateInboxRulesResponse>
</s:Body>
</s:Envelope>

```

4.2 SetInboxRule

The following is an example of an **UpdateInboxRules** operation that updates an Inbox rule (2).

The client constructs the request XML and sends it to the server. The **SetRuleOperation** complex type contains the rule (2) to be updated. This example updates the existing identified rule (2), which will search for a subject string "This is Junk". The **FolderId** attribute has been shortened to preserve readability.

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010_SP1" />
  </soap:Header>
  <soap:Body>
    <m:UpdateInboxRules>
      <m:RemoveOutlookRuleBlob>true</m:RemoveOutlookRuleBlob>
      <m:Operations>
        <t:SetRuleOperation>
          <t:Rule>
            <t:RuleId>dCsAAABjzwQ=</t:RuleId>
            <t:DisplayName>MoveInterestingToJunk</t:DisplayName>
            <t:Priority>1</t:Priority>
            <t:IsEnabled>true</t:IsEnabled>
            <t:IsInError>>false</t:IsInError>
            <t:Conditions>
              <t:ContainsSubjectStrings>
                <t:String>This is Junk</t:String>
              </t:ContainsSubjectStrings>
            </t:Conditions>
            <t:Exceptions />
            <t:Actions>
              <t:MoveToFolder>
                <t:FolderId Id="AAMkAGYzZjZm=" ChangeKey="AQAAAA==" />
              </t:MoveToFolder>
            </t:Actions>
          </t:Rule>
        </t:SetRuleOperation>
      </m:Operations>
    </m:UpdateInboxRules>
  </soap:Body>
</soap:Envelope>

```

The server constructs the response XML and sends it to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
      MinorVersion="1"
      MajorBuildNumber="139"
      MinorBuildNumber="0"
      Version="Exchange2010_SP1"
      xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <UpdateInboxRulesResponse ResponseClass="Success"
      xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <ResponseCode>NoError</ResponseCode>
    </UpdateInboxRulesResponse>
  </s:Body>
</s:Envelope>
```

4.3 DeleteInboxRule

The following is an example of an **UpdateInboxRules** operation that deletes an Inbox rule (2).

The client constructs the request XML and sends it to the server. The **DeleteRuleOperation** complex type contains the **RuleId** element, which contains the rule (2) ID of the rule (2) to be deleted.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010_SP1" />
  </soap:Header>
  <soap:Body>
    <m:UpdateInboxRules>
      <m:RemoveOutlookRuleBlob>true</m:RemoveOutlookRuleBlob>
      <m:Operations>
        <t>DeleteRuleOperation>
          <t:RuleId>dCsAAABjzwY</t:RuleId>
        </t>DeleteRuleOperation>
      </m:Operations>
    </m:UpdateInboxRules>
  </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
      MinorVersion="1" MajorBuildNumber="139"
      MinorBuildNumber="0"
      Version="Exchange2010_SP1"
      xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
```

```

        xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
</s:Header>
<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <UpdateInboxRulesResponse ResponseClass="Success"
    xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
    <ResponseCode>NoError</ResponseCode>
  </UpdateInboxRulesResponse>
</s:Body>
</s:Envelope>

```

4.4 GetInboxRules

The following is an example of a **GetInboxRules** operation.

The client constructs the request XML and sends it to the server. The request identifies the user in the **MailboxSmtpAddress** element. All Inbox rules (2) for the identified user are to be returned in the response.

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010_SP1" />
  </soap:Header>
  <soap:Body>
    <m:GetInboxRules>
      <m:MailboxSmtpAddress>User1@Contoso.com</m:MailboxSmtpAddress>
    </m:GetInboxRules>
  </soap:Body>
</soap:Envelope>

```

The server constructs the response XML and sends it to the client. The **FolderId** attribute has been shortened to preserve readability. In this example, there is one rule (2) in the returned **InboxRules** collection.

```

<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
      MinorVersion="1" MajorBuildNumber="139"
      MinorBuildNumber="0"
      Version="Exchange2010_SP1"
      xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <GetInboxRulesResponse ResponseClass="Success"
      xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <ResponseCode>NoError</ResponseCode>
      <OutlookRuleBlobExists>true</OutlookRuleBlobExists>
      <InboxRules>
        <Rule xmlns="http://schemas.microsoft.com/exchange/services/2006/types">
          <RuleId>dCsAAABjzvA</RuleId>

```

```

    <DisplayName>MoveInterestingToJunk</DisplayName>
    <Priority>1</Priority>
    <IsEnabled>true</IsEnabled>
    <Conditions>
      <ContainsSubjectStrings>
        <String>Interesting</String>
      </ContainsSubjectStrings>
    </Conditions>
    <Actions>
      <MoveToFolder>
        <FolderId ChangeKey="AQAAAA==" Id="AAMkAGYzZjZm" />
      </MoveToFolder>
    </Actions>
  </Rule>
</InboxRules>
</GetInboxRulesResponse>
</s:Body>
</s:Envelope>

```

4.5 RuleOperationErrorType

The following is an example of an **UpdateInboxRules** operation that results in a validation error. This example sets the **RuleId** element, which specifies the rule (2) identifier. When creating a rule (2), the **Rule** element cannot include a **RuleID** element. The rule (2) identifier is generated when the rule (2) is successfully created in the server message store.

The client constructs the request XML and sends it to the server. The **CreateRuleOperation** complex type contains the **Rule** complex type to be created. This example defines a **RuleId** element, which causes the error.

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010 SP1" />
  </soap:Header>
  <soap:Body>
    <m:UpdateInboxRules>
      <m:RemoveOutlookRuleBlob>true</m:RemoveOutlookRuleBlob>
      <m:Operations>
        <t:CreateRuleOperation>
          <t:Rule>
            <t:RuleId>dCsAAABjz0Q=</t:RuleId>
            <t:DisplayName>MoveInterestingToJunk</t:DisplayName>
            <t:Priority>1</t:Priority>
            <t:IsEnabled>true</t:IsEnabled>
            <t:IsInError>>false</t:IsInError>
            <t:Conditions>
              <t:ContainsSubjectStrings>
                <t:String>Interesting</t:String>
              </t:ContainsSubjectStrings>
            </t:Conditions>
            <t:Exceptions />
            <t:Actions>
              <t:MoveToFolder>
                <t:DistinguishedFolderId Id="inbox" />
              </t:MoveToFolder>
            </t:Actions>
          </t:Rule>
        </t:CreateRuleOperation>
      </m:Operations>
    </m:UpdateInboxRules>
  </soap:Body>
</soap:Envelope>

```

```

    </m:UpdateInboxRules>
  </soap:Body>
</soap:Envelope>

```

The server constructs the response XML and sends it to the client. The **RuleValidationError** complex type contains the information about the cause of the validation error.

```

<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
      MinorVersion="1"
      MajorBuildNumber="139"
      MinorBuildNumber="0"
      Version="Exchange2010_SP1"
      xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <UpdateInboxRulesResponse ResponseClass="Error"
      xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <MessageText>Validation error occurred during rule operation execution.</MessageText>
      <ResponseCode>ErrorInboxRulesValidationError</ResponseCode>
      <DescriptiveLinkKey>0</DescriptiveLinkKey>
      <RuleOperationErrors>
        <RuleOperationError
          xmlns="http://schemas.microsoft.com/exchange/services/2006/types">
          <OperationIndex>0</OperationIndex>
          <ValidationErrors>
            <Error>
              <FieldURI>RuleId</FieldURI>
              <ErrorCode>CreateWithRuleId</ErrorCode>
              <ErrorMessage>Create rule operation should not have rule ID set in
request.</ErrorMessage>
              <FieldValue>dCsAAABjz0Q=</FieldValue>
            </Error>
          </ValidationErrors>
        </RuleOperationError>
      </RuleOperationErrors>
    </UpdateInboxRulesResponse>
  </s:Body>
</s:Envelope>

```

4.6 RuleValidationErrorType

The following is an example of an **UpdateInboxRules** operation that results in a validation error. This example creates a **Rule** element, which specifies a **MaximumSize** of 9999990, which fails validation.

The client constructs the request XML and sends it to the server. The **CreateRuleOperation** complex type contains the **Rule** complex type to be created. This example defines an improper **Rule** element, which causes the error.

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">

```



```

xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
<soap:Header>
  <t:RequestServerVersion Version="Exchange2010_SP1" />
</soap:Header>
<soap:Body>
  <m:UpdateInboxRules>
    <m:RemoveOutlookRuleBlob>true</m:RemoveOutlookRuleBlob>
    <m:Operations>
      <t:CreateRuleOperation>
        <t:Rule>
          <t:DisplayName>MoveInterestingToJunk</t:DisplayName>
          <t:Priority>1</t:Priority>
          <t:IsEnabled>true</t:IsEnabled>
          <t:IsInError>>false</t:IsInError>
          <t:Conditions>
            <t:ContainsSubjectStrings>
              <t:String>Interesting</t:String>
            </t:ContainsSubjectStrings>
            <t:WithinDateRange>
              <t:StartDateTime>2011-04-26T18:05:13Z</t:StartDateTime>
              <t:EndDateTime>2011-04-26T20:05:13Z</t:EndDateTime>
            </t:WithinDateRange>
            <t:WithinSizeRange>
              <t:MinimumSize>0</t:MinimumSize>
              <t:MaximumSize>9999990</t:MaximumSize>
            </t:WithinSizeRange>
          </t:Conditions>
          <t:Exceptions />
          <t:Actions>
            <t:MoveToFolder>
              <t:DistinguishedFolderId Id="junkemail" />
            </t:MoveToFolder>
          </t:Actions>
        </t:Rule>
      </t:CreateRuleOperation>
    </m:Operations>
  </m:UpdateInboxRules>
</soap:Body>
</soap:Envelope>

```

The server constructs the response XML and sends it to the client. The **RuleValidationError** complex type contains the **Error** information about the cause of the validation error. In this example, the **FieldValue** of 9999990 and has an **ErrorCode** stating that this is **InvalidValue**.

```

<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14" MinorVersion="1" MajorBuildNumber="225"
      MinorBuildNumber="42" Version="Exchange2010 SP1"
      xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <UpdateInboxRulesResponse ResponseClass="Error"
      xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <MessageText>A validation error occurred while executing the rule
        operation.</MessageText>
      <ResponseCode>ErrorInboxRulesValidationError</ResponseCode>
      <DescriptiveLinkKey>0</DescriptiveLinkKey>
      <RuleOperationErrors>
        <RuleOperationError
          xmlns="http://schemas.microsoft.com/exchange/services/2006/types">

```

```
<OperationIndex>0</OperationIndex>
<ValidationErrors>
  <Error>
    <FieldURI>Condition:WithinSizeRange</FieldURI>
    <ErrorCode>InvalidValue</ErrorCode>
    <ErrorMessage>The value isn't valid.</ErrorMessage>
    <FieldValue>9999990</FieldValue>
  </Error>
</ValidationErrors>
</RuleOperationError>
</RuleOperationErrors>
</UpdateInboxRulesResponse>
</s:Body>
</s:Envelope>
```

Preliminary

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

Preliminary

6 Appendix A: Full WSDL

The XML files that are listed in the following table are required in order to implement the functionality described in this document.

File name	Description	Section
MS-OXWSRULES.wsdl	Contains the WSDL for the implementation of this protocol.	6
MS-OXWSRULES-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	7.1
MS-OXWSRULES-types.xsd	Contains the XML schema type definitions that are used in this protocol.	7.2

These files have to be placed in a common folder for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSRULES-types.xsd or MS-OXWSRULES-messages.xsd schemas have to be placed in the common folder with these files.

This section contains the contents of the MS-OXWSRULES.wsdl file.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:s="http://www.w3.org/2001/XMLSchema"

  targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/messages"
        schemaLocation="MS-OXWSRULES-messages.xsd"/>
    </xs:schema>
  </wsdl:types>

  <!-- Inbox Rules Messages -->
  <wsdl:message name="GetInboxRulesSoapIn">
    <wsdl:part name="request" element="tns:GetInboxRules" />
    <wsdl:part name="Impersonation" element="t:ExchangeImpersonation" />
    <wsdl:part name="MailboxCulture" element="t:MailboxCulture" />
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion" />
    <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext" />
  </wsdl:message>
  <wsdl:message name="GetInboxRulesSoapOut">
    <wsdl:part name="GetInboxRulesResult" element="tns:GetInboxRulesResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo" />
  </wsdl:message>

  <wsdl:message name="UpdateInboxRulesSoapIn">
    <wsdl:part name="request" element="tns:UpdateInboxRules" />
    <wsdl:part name="Impersonation" element="t:ExchangeImpersonation" />
    <wsdl:part name="MailboxCulture" element="t:MailboxCulture" />
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion" />
    <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext" />
  </wsdl:message>
  <wsdl:message name="UpdateInboxRulesSoapOut">
    <wsdl:part name="UpdateInboxRulesResult" element="tns:UpdateInboxRulesResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo" />
  </wsdl:message>

  <wsdl:portType name="ExchangeServicePortType">
    <!-- Inbox Rules -->
```

```

    <wsdl:operation name="GetInboxRules">
      <wsdl:input message="tns:GetInboxRulesSoapIn" />
      <wsdl:output message="tns:GetInboxRulesSoapOut" />
    </wsdl:operation>

    <wsdl:operation name="UpdateInboxRules">
      <wsdl:input message="tns:UpdateInboxRulesSoapIn" />
      <wsdl:output message="tns:UpdateInboxRulesSoapOut" />
    </wsdl:operation>
  </wsdl:portType>

  <wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
    <wsdl:documentation>
      <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0" xmlns:wsi="http://ws-
i.org/schemas/conformanceClaim/" />
    </wsdl:documentation>
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />

    <!-- Inbox Rules Operations -->

    <!-- GetInboxRules -->
    <wsdl:operation name="GetInboxRules">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetInboxRules" />
      <wsdl:input>
        <soap:header message="tns:GetInboxRulesSoapIn" part="Impersonation" use="literal" />
        <soap:header message="tns:GetInboxRulesSoapIn" part="MailboxCulture" use="literal" />
        <soap:header message="tns:GetInboxRulesSoapIn" part="RequestVersion" use="literal" />
        <soap:header message="tns:GetInboxRulesSoapIn" part="TimeZoneContext" use="literal"
/>
        <soap:body parts="request" use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="GetInboxRulesResult" use="literal" />
        <soap:header message="tns:GetInboxRulesSoapOut" part="ServerVersion" use="literal" />
      </wsdl:output>
    </wsdl:operation>

    <!-- UpdateInboxRules -->
    <wsdl:operation name="UpdateInboxRules">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/UpdateInboxRules" />
      <wsdl:input>
        <soap:header message="tns:UpdateInboxRulesSoapIn" part="Impersonation" use="literal"
/>
        <soap:header message="tns:UpdateInboxRulesSoapIn" part="MailboxCulture" use="literal"
/>
        <soap:header message="tns:UpdateInboxRulesSoapIn" part="RequestVersion" use="literal"
/>
        <soap:header message="tns:UpdateInboxRulesSoapIn" part="TimeZoneContext"
use="literal" />
        <soap:body parts="request" use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="UpdateInboxRulesResult" use="literal" />
        <soap:header message="tns:UpdateInboxRulesSoapOut" part="ServerVersion" use="literal"
/>
      </wsdl:output>
    </wsdl:operation>

  </wsdl:binding>
</wsdl:definitions>

```

7 Appendix B: Full XML Schema

For ease of implementation, the following sections provide the full XML schema for this protocol.

Schema name	Prefix	Section
Messages schema	m:	7.1
Types schema	t:	7.2

These files have to be placed in a common folder in order for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSRULES-types.xsd or MS-OXWSRULES-messages.xsd schemas have to be placed in the common folder along with the files listed in the table.

7.1 Messages Schema

This section contains the contents of the MS-OXWSRULES-messages.xsd file and information about additional files that this schema file requires to operate correctly.

MS-OXWSRULES-types.xsd includes the file listed in the following table. For the schema file to operate correctly, this file has to be in the folder that contains the WSDL, types schema, and messages schema files for this protocol.

File name	Defining specification
MS-OXWSCDATA -messages.xsd	[MS-OXWSCDATA] section 7.1

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
elementFormDefault="qualified" version="Exchange2016" id="messages">

  <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="MS-OXWSRULES-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>

  <xs:element name="GetInboxRules" type="m:GetInboxRulesRequestType"/>

  <xs:complexType name="GetInboxRulesRequestType">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:element name="MailboxSmtAddress" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="GetInboxRulesResponse" type="m:GetInboxRulesResponseType"/>

  <xs:complexType name="GetInboxRulesResponseType">
    <xs:complexContent>
      <xs:extension base="m:ResponseMessageType">
        <xs:sequence>
          <xs:element name="OutlookRuleBlobExists" type="xs:boolean" minOccurs="0"/>
          <xs:element name="InboxRules" type="t:ArrayOfRulesType" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

```

    </xs:complexContent>
  </xs:complexType>

  <xs:element name="UpdateInboxRules" type="m:UpdateInboxRulesRequestType"/>

  <xs:complexType name="UpdateInboxRulesRequestType">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:element name="MailboxSmtAddress" type="xs:string" minOccurs="0"/>
          <xs:element name="RemoveOutlookRuleBlob" type="xs:boolean" minOccurs="0"/>
          <xs:element name="Operations" type="t:ArrayOfRuleOperationsType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="UpdateInboxRulesResponse" type="m:UpdateInboxRulesResponseType"/>

  <xs:complexType name="UpdateInboxRulesResponseType">
    <xs:complexContent>
      <xs:extension base="m:ResponseMessageType">
        <xs:sequence>
          <xs:element name="RuleOperationErrors" type="t:ArrayOfRuleOperationErrorsType"
minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>

```

7.2 Types Schema

This section contains the contents of the MS-OXWSRULES-types.xsd file and information about additional files that this schema file requires to operate correctly.

MS-OXWSRULES-types.xsd includes the files listed in the following table. For the schema file to operate correctly, these files have to be present in the folder that contains the WSDL, types schema, and messages schema files for this protocol.

File name	Defining specification
MS-OXWSCDATA-types.xsd	[MS-OXWSCDATA] section 7.2
MS-OXWSCORE-types.xsd	[MS-OXWSCORE] section 7.2

```

<?xml version="1.0" encoding="utf-8"?>

<xs:schema xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
elementFormDefault="qualified" version="Exchange2016" id="types">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSCORE-types.xsd"/>

  <xs:complexType name="ArrayOfRulesType">
    <xs:sequence>
      <xs:element name="Rule" type="t:RuleType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="RuleType">

```

```

<xs:sequence>
  <xs:element name="RuleId" type="xs:string" minOccurs="0"/>
  <xs:element name="DisplayName" type="xs:string"/>
  <xs:element name="Priority" type="xs:int"/>
  <xs:element name="IsEnabled" type="xs:boolean"/>
  <xs:element name="IsNotSupported" type="xs:boolean" minOccurs="0"/>
  <xs:element name="IsInError" type="xs:boolean" minOccurs="0"/>
  <xs:element name="Conditions" type="t:RulePredicatesType" minOccurs="0"/>
  <xs:element name="Exceptions" type="t:RulePredicatesType" minOccurs="0"/>
  <xs:element name="Actions" type="t:RuleActionsType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="RulePredicatesType">
  <xs:sequence>
    <xs:element name="Categories" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="ContainsBodyStrings" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="ContainsHeaderStrings" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="ContainsRecipientStrings" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="ContainsSenderStrings" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="ContainsSubjectOrBodyStrings" type="t:ArrayOfStringsType"
minOccurs="0"/>
    <xs:element name="ContainsSubjectStrings" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="FlaggedForAction" type="t:FlaggedForActionType" minOccurs="0"/>
    <xs:element name="FromAddresses" type="t:ArrayOfEmailAddressesType" minOccurs="0"/>
    <xs:element name="FromConnectedAccounts" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="HasAttachments" type="xs:boolean" minOccurs="0"/>
    <xs:element name="Importance" type="t:ImportanceChoicesType" minOccurs="0"/>
    <xs:element name="IsApprovalRequest" type="xs:boolean" minOccurs="0"/>
    <xs:element name="IsAutomaticForward" type="xs:boolean" minOccurs="0"/>
    <xs:element name="IsAutomaticReply" type="xs:boolean" minOccurs="0"/>
    <xs:element name="IsEncrypted" type="xs:boolean" minOccurs="0"/>
    <xs:element name="IsMeetingRequest" type="xs:boolean" minOccurs="0"/>
    <xs:element name="IsMeetingResponse" type="xs:boolean" minOccurs="0"/>
    <xs:element name="IsNDR" type="xs:boolean" minOccurs="0"/>
    <xs:element name="IsPermissionControlled" type="xs:boolean" minOccurs="0"/>
    <xs:element name="IsReadReceipt" type="xs:boolean" minOccurs="0"/>
    <xs:element name="IsSigned" type="xs:boolean" minOccurs="0"/>
    <xs:element name="IsVoicemail" type="xs:boolean" minOccurs="0"/>
    <xs:element name="ItemClasses" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="MessageClassifications" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="NotSentToMe" type="xs:boolean" minOccurs="0"/>
    <xs:element name="SentCcMe" type="xs:boolean" minOccurs="0"/>
    <xs:element name="SentOnlyToMe" type="xs:boolean" minOccurs="0"/>
    <xs:element name="SentToAddresses" type="t:ArrayOfEmailAddressesType" minOccurs="0"/>
    <xs:element name="SentToMe" type="xs:boolean" minOccurs="0"/>
    <xs:element name="SentToOrCcMe" type="xs:boolean" minOccurs="0"/>
    <xs:element name="Sensitivity" type="t:SensitivityChoicesType" minOccurs="0"/>
    <xs:element name="WithinDateRange" type="t:RulePredicateDateRangeType" minOccurs="0"/>
    <xs:element name="WithinSizeRange" type="t:RulePredicateSizeRangeType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RuleActionsType">
  <xs:sequence>
    <xs:element name="AssignCategories" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="CopyToFolder" type="t:TargetFolderIdType" minOccurs="0"/>
    <xs:element name="Delete" type="xs:boolean" minOccurs="0"/>
    <xs:element name="ForwardAsAttachmentToRecipients" type="t:ArrayOfEmailAddressesType"
minOccurs="0"/>
    <xs:element name="ForwardToRecipients" type="t:ArrayOfEmailAddressesType"
minOccurs="0"/>
    <xs:element name="MarkImportance" type="t:ImportanceChoicesType" minOccurs="0"/>
    <xs:element name="MarkAsRead" type="xs:boolean" minOccurs="0"/>
    <xs:element name="MoveToFolder" type="t:TargetFolderIdType" minOccurs="0"/>
    <xs:element name="PermanentDelete" type="xs:boolean" minOccurs="0"/>
    <xs:element name="RedirectToRecipients" type="t:ArrayOfEmailAddressesType"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

```



```

    <xs:element name="SendSMSAlertToRecipients" type="t:ArrayOfEmailAddressesType"
minOccurs="0"/>
    <xs:element name="ServerReplyWithMessage" type="t:ItemIdType" minOccurs="0"/>
    <xs:element name="StopProcessingRules" type="xs:boolean" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="FlaggedForActionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Any"/>
    <xs:enumeration value="Call"/>
    <xs:enumeration value="DoNotForward"/>
    <xs:enumeration value="FollowUp"/>
    <xs:enumeration value="FYI"/>
    <xs:enumeration value="Forward"/>
    <xs:enumeration value="NoResponseNecessary"/>
    <xs:enumeration value="Read"/>
    <xs:enumeration value="Reply"/>
    <xs:enumeration value="ReplyToAll"/>
    <xs:enumeration value="Review"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="RulePredicateDateRangeType">
  <xs:sequence>
    <xs:element name="StartDateTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="EndDateTime" type="xs:dateTime" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RulePredicateSizeRangeType">
  <xs:sequence>
    <xs:element name="MinimumSize" type="xs:int" minOccurs="0"/>
    <xs:element name="MaximumSize" type="xs:int" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ArrayOfRuleOperationsType">
  <xs:choice maxOccurs="unbounded">
    <xs:element name="CreateRuleOperation" type="t:CreateRuleOperationType"/>
    <xs:element name="SetRuleOperation" type="t:SetRuleOperationType"/>
    <xs:element name="DeleteRuleOperation" type="t>DeleteRuleOperationType"/>
  </xs:choice>
</xs:complexType>

<xs:complexType name="CreateRuleOperationType">
  <xs:complexContent>
    <xs:extension base="t:RuleOperationType">
      <xs:sequence>
        <xs:element name="Rule" type="t:RuleType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="RuleOperationType" abstract="true">
</xs:complexType>

<xs:complexType name="SetRuleOperationType">
  <xs:complexContent>
    <xs:extension base="t:RuleOperationType">
      <xs:sequence>
        <xs:element name="Rule" type="t:RuleType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

</xs:complexType>

<xs:complexType name="DeleteRuleOperationType">
  <xs:complexContent>
    <xs:extension base="t:RuleOperationType">
      <xs:sequence>
        <xs:element name="RuleId" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="ArrayOfRuleOperationErrorsType">
  <xs:sequence>
    <xs:element name="RuleOperationError" type="t:RuleOperationErrorType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RuleOperationErrorType">
  <xs:sequence>
    <xs:element name="OperationIndex" type="xs:int"/>
    <xs:element name="ValidationErrors" type="t:ArrayOfRuleValidationErrorsType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ArrayOfRuleValidationErrorsType">
  <xs:sequence>
    <xs:element name="Error" type="t:RuleValidationErrorType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="RuleValidationErrorType">
  <xs:sequence>
    <xs:element name="FieldURI" type="t:RuleFieldURIType"/>
    <xs:element name="ErrorCode" type="t:RuleValidationErrorCodeType"/>
    <xs:element name="ErrorMessage" type="xs:string"/>
    <xs:element name="FieldValue" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="RuleFieldURIType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="RuleId"/>
    <xs:enumeration value="DisplayName"/>
    <xs:enumeration value="Priority"/>
    <xs:enumeration value="IsNotSupported"/>
    <xs:enumeration value="Actions"/>
    <xs:enumeration value="Condition:Categories"/>
    <xs:enumeration value="Condition:ContainsBodyStrings"/>
    <xs:enumeration value="Condition:ContainsHeaderStrings"/>
    <xs:enumeration value="Condition:ContainsRecipientStrings"/>
    <xs:enumeration value="Condition:ContainsSenderStrings"/>
    <xs:enumeration value="Condition:ContainsSubjectOrBodyStrings"/>
    <xs:enumeration value="Condition:ContainsSubjectStrings"/>
    <xs:enumeration value="Condition:FlaggedForAction"/>
    <xs:enumeration value="Condition:FromAddresses"/>
    <xs:enumeration value="Condition:FromConnectedAccounts"/>
    <xs:enumeration value="Condition:HasAttachments"/>
    <xs:enumeration value="Condition:Importance"/>
    <xs:enumeration value="Condition:IsApprovalRequest"/>
    <xs:enumeration value="Condition:IsAutomaticForward"/>
    <xs:enumeration value="Condition:IsAutomaticReply"/>
    <xs:enumeration value="Condition:IsEncrypted"/>
    <xs:enumeration value="Condition:IsMeetingRequest"/>
    <xs:enumeration value="Condition:IsMeetingResponse"/>
    <xs:enumeration value="Condition:IsNDR"/>
    <xs:enumeration value="Condition:IsPermissionControlled"/>
  </xs:restriction>
</xs:simpleType>

```

```

<xs:enumeration value="Condition:IsReadReceipt"/>
<xs:enumeration value="Condition:IsSigned"/>
<xs:enumeration value="Condition:IsVoicemail"/>
<xs:enumeration value="Condition:ItemClasses"/>
<xs:enumeration value="Condition:MessageClassifications"/>
<xs:enumeration value="Condition:NotSentToMe"/>
<xs:enumeration value="Condition:SentCcMe"/>
<xs:enumeration value="Condition:SentOnlyToMe"/>
<xs:enumeration value="Condition:SentToAddresses"/>
<xs:enumeration value="Condition:SentToMe"/>
<xs:enumeration value="Condition:SentToOrCcMe"/>
<xs:enumeration value="Condition:Sensitivity"/>
<xs:enumeration value="Condition:WithinDateRange"/>
<xs:enumeration value="Condition:WithinSizeRange"/>
<xs:enumeration value="Exception:Categories"/>
<xs:enumeration value="Exception:ContainsBodyStrings"/>
<xs:enumeration value="Exception:ContainsHeaderStrings"/>
<xs:enumeration value="Exception:ContainsRecipientStrings"/>
<xs:enumeration value="Exception:ContainsSenderStrings"/>
<xs:enumeration value="Exception:ContainsSubjectOrBodyStrings"/>
<xs:enumeration value="Exception:ContainsSubjectStrings"/>
<xs:enumeration value="Exception:FlaggedForAction"/>
<xs:enumeration value="Exception:FromAddresses"/>
<xs:enumeration value="Exception:FromConnectedAccounts"/>
<xs:enumeration value="Exception:HasAttachments"/>
<xs:enumeration value="Exception:Importance"/>
<xs:enumeration value="Exception:IsApprovalRequest"/>
<xs:enumeration value="Exception:IsAutomaticForward"/>
<xs:enumeration value="Exception:IsAutomaticReply"/>
<xs:enumeration value="Exception:IsEncrypted"/>
<xs:enumeration value="Exception:IsMeetingRequest"/>
<xs:enumeration value="Exception:IsMeetingResponse"/>
<xs:enumeration value="Exception:IsNDR"/>
<xs:enumeration value="Exception:IsPermissionControlled"/>
<xs:enumeration value="Exception:IsReadReceipt"/>
<xs:enumeration value="Exception:IsSigned"/>
<xs:enumeration value="Exception:IsVoicemail"/>
<xs:enumeration value="Exception:ItemClasses"/>
<xs:enumeration value="Exception:MessageClassifications"/>
<xs:enumeration value="Exception:NotSentToMe"/>
<xs:enumeration value="Exception:SentCcMe"/>
<xs:enumeration value="Exception:SentOnlyToMe"/>
<xs:enumeration value="Exception:SentToAddresses"/>
<xs:enumeration value="Exception:SentToMe"/>
<xs:enumeration value="Exception:SentToOrCcMe"/>
<xs:enumeration value="Exception:Sensitivity"/>
<xs:enumeration value="Exception:WithinDateRange"/>
<xs:enumeration value="Exception:WithinSizeRange"/>
<xs:enumeration value="Action:AssignCategories"/>
<xs:enumeration value="Action:CopyToFolder"/>
<xs:enumeration value="Action>Delete"/>
<xs:enumeration value="Action:ForwardAsAttachmentToRecipients"/>
<xs:enumeration value="Action:ForwardToRecipients"/>
<xs:enumeration value="Action:MarkImportance"/>
<xs:enumeration value="Action:MarkAsRead"/>
<xs:enumeration value="Action:MoveToFolder"/>
<xs:enumeration value="Action:PermanentDelete"/>
<xs:enumeration value="Action:RedirectToRecipients"/>
<xs:enumeration value="Action:SendSMSAlertToRecipients"/>
<xs:enumeration value="Action:ServerReplyWithMessage"/>
<xs:enumeration value="Action:StopProcessingRules"/>
<xs:enumeration value="IsEnabled"/>
<xs:enumeration value="IsInError"/>
<xs:enumeration value="Conditions"/>
<xs:enumeration value="Exceptions"/>
</xs:restriction>
</xs:simpleType>

```

```
<xs:simpleType name="RuleValidationErrorCodeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ADOperationFailure"/>
    <xs:enumeration value="ConnectedAccountNotFound"/>
    <xs:enumeration value="CreateWithRuleId"/>
    <xs:enumeration value="EmptyValueFound"/>
    <xs:enumeration value="DuplicatedPriority"/>
    <xs:enumeration value="DuplicatedOperationOnTheSameRule"/>
    <xs:enumeration value="FolderDoesNotExist"/>
    <xs:enumeration value="InvalidAddress"/>
    <xs:enumeration value="InvalidDateRange"/>
    <xs:enumeration value="InvalidFolderId"/>
    <xs:enumeration value="InvalidSizeRange"/>
    <xs:enumeration value="InvalidValue"/>
    <xs:enumeration value="MessageClassificationNotFound"/>
    <xs:enumeration value="MissingAction"/>
    <xs:enumeration value="MissingParameter"/>
    <xs:enumeration value="MissingRangeValue"/>
    <xs:enumeration value="NotSettable"/>
    <xs:enumeration value="RecipientDoesNotExist"/>
    <xs:enumeration value="RuleNotFound"/>
    <xs:enumeration value="SizeLessThanZero"/>
    <xs:enumeration value="StringValueTooBig"/>
    <xs:enumeration value="UnsupportedAddress"/>
    <xs:enumeration value="UnexpectedError"/>
    <xs:enumeration value="UnsupportedRule"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

Preliminary

8 Appendix C: 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 2010 Service Pack 1 (SP1)
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 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.

Preliminary

9 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
8 Appendix C: Product Behavior	Added Exchange 2016 to list of supported products.	Y	Content update.

Preliminary

10 Index

A

[Applicability](#) 10
[Attribute groups](#) 24
[Attributes](#) 24

C

[Capability negotiation](#) 10
[Change tracking](#) 70
[Complex types](#) 12
[t:RuleActionTypes Complex Type](#) 12
[t:RulePredicateDateRangeType Complex Type](#) 14
[t:RulePredicateSizeRangeType Complex Type](#) 15
[t:RulePredicatesType Complex Type](#) 15
[t:RuleType Complex Type](#) 21

CreateInboxRule example
Examples
[CreateInboxRule](#) 51

D

DeleteInboxRule example
Examples
[DeleteInboxRule](#) 53

E

Examples
[overview](#) 51

F

[Fields - vendor-extensible](#) 10
[Full WSDL](#) 60
[Full XML schema](#) 62
[Messages Schema](#) 62
[Types Schema](#) 63

G

GetInboxRules example
Examples
[GetInboxRules](#) 54
[Glossary](#) 6
[Groups](#) 24

I

[Implementer - security considerations](#) 59
[Index of security parameters](#) 59
[Informative references](#) 9
[Introduction](#) 6

M

Messages
[attribute groups](#) 24
[attributes](#) 24

[complex types](#) 12
[elements](#) 11
[enumerated](#) 11
[groups](#) 24
[namespaces](#) 11
[simple types](#) 23
[syntax](#) 11
[t:FlaggedForActionType Simple Type simple type](#) 23
[t:RuleActionTypes Complex Type complex type](#) 12
[t:RulePredicateDateRangeType Complex Type complex type](#) 14
[t:RulePredicateSizeRangeType Complex Type complex type](#) 15
[t:RulePredicatesType Complex Type complex type](#) 15
[t:RuleType Complex Type complex type](#) 21
[transport](#) 11

N

[Namespaces](#) 11
[Normative references](#) 8

O

[Overview \(synopsis\)](#) 9

P

[Parameters - security index](#) 59
[Preconditions](#) 9
[Prerequisites](#) 9
[Product behavior](#) 69
Protocol Details
[overview](#) 25

R

[References](#) 8
[informative](#) 9
[normative](#) 8
[Relationship to other protocols](#) 9
RuleOperationErrorType example
Examples
[RuleOperationErrorType](#) 55
RuleValidationErrorType example
Examples
[RuleValidationErrorType](#) 56

S

Security
[implementer considerations](#) 59
[parameter index](#) 59
SetInboxRule example
Examples
[SetInboxRule](#) 52
[Simple types](#) 23
[t:FlaggedForActionType Simple Type](#) 23

[Standards assignments](#) 10
Syntax
[messages - overview](#) 11

T

[t:FlaggedForActionType Simple Type simple type](#) 23
[t:RuleActionsType Complex Type complex type](#) 12
[t:RulePredicateDateRangeType Complex Type complex type](#) 14
[t:RulePredicateSizeRangeType Complex Type complex type](#) 15
[t:RulePredicatesType Complex Type complex type](#) 15
[t:RuleType Complex Type complex type](#) 21
[Tracking changes](#) 70
[Transport](#) 11
Types
[complex](#) 12
[simple](#) 23

V

[Vendor-extensible fields](#) 10
[Versioning](#) 10

W

[WSDL](#) 60

X

[XML schema](#) 62
[Messages Schema](#) 62
[Types Schema](#) 63

Preliminary