Inbox Rules Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might 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](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact dochelp@microsoft.com.
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/5/2010</td>
<td>0.1.0</td>
<td>Major</td>
<td>Initial Release.</td>
</tr>
<tr>
<td>8/4/2010</td>
<td>1.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>11/3/2010</td>
<td>2.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>3.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>8/5/2011</td>
<td>4.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/7/2011</td>
<td>4.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>1/20/2012</td>
<td>5.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/27/2012</td>
<td>5.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>5.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>5.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>5.2</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/26/2013</td>
<td>5.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>5.4</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>5.4</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>5.5</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>5.6</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>5.7</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>5/26/2015</td>
<td>6.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>9/14/2015</td>
<td>6.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/13/2016</td>
<td>6.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>6.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/20/2017</td>
<td>6.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/24/2018</td>
<td>7.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/1/2018</td>
<td>8.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>2/15/2022</td>
<td>8.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
</tbody>
</table>
Table of Contents

1 Introduction ........................................................................................................... 5
  1.1 Glossary ........................................................................................................... 5
  1.2 References ...................................................................................................... 7
    1.2.1 Normative References .......................................................................... 7
    1.2.2 Informative References ......................................................................... 7
  1.3 Overview .......................................................................................................... 8
  1.4 Relationship to Other Protocols .................................................................... 8
  1.5 Prerequisites/Preconditions ........................................................................ 8
  1.6 Applicability Statement ................................................................................ 8
  1.7 Versioning and Capability Negotiation ....................................................... 8
  1.8 Vendor-Extensible Fields ............................................................................. 9
  1.9 Standards Assignments .............................................................................. 9

2 Messages................................................................................................................ 10
  2.1 Transport ........................................................................................................ 10
  2.2 Common Message Syntax .......................................................................... 10
    2.2.1 Namespaces ............................................................................................ 10
    2.2.2 Messages ................................................................................................ 10
    2.2.3 Elements ................................................................................................ 10
    2.2.4 Complex Types ....................................................................................... 11
      2.2.4.1 t:RuleActionsType Complex Type .................................................. 11
      2.2.4.2 t:RulePredicateDateRangeType Complex Type .......................... 13
      2.2.4.3 t:RulePredicateSizeRangeType Complex Type ......................... 14
      2.2.4.4 t:RulePredicatesType Complex Type .......................................... 14
      2.2.4.5 t:RuleType Complex Type .............................................................. 20
    2.2.5 Simple Types ............................................................................................ 21
      2.2.5.1 t:FlaggedForActionType Simple Type .......................................... 22
    2.2.6 Attributes ................................................................................................ 23
    2.2.7 Groups ...................................................................................................... 23
    2.2.8 Attribute Groups ..................................................................................... 23

3 Protocol Details ..................................................................................................... 24
  3.1 ExchangeServicePortType Server Detail .................................................... 24
    3.1.1 Abstract Data Model ............................................................................. 24
    3.1.2 Timers ..................................................................................................... 24
    3.1.3 Initialization ............................................................................................ 24
    3.1.4 Message Processing Events and Sequencing Rules ......................... 24
      3.1.4.1 GetInboxRules .................................................................................. 24
        3.1.4.1.1 Messages ................................................................................... 25
          3.1.4.1.1.1 tns:GetInboxRulesSoapIn Message .................................... 25
          3.1.4.1.1.2 tns:GetInboxRulesSoapOut Message .................................. 26
        3.1.4.1.2 Elements .................................................................................... 26
          3.1.4.1.2.1 GetInboxRules Element ...................................................... 26
          3.1.4.1.2.2 GetInboxRulesResponse Element ..................................... 27
        3.1.4.1.3 Complex Types .......................................................................... 27
          3.1.4.1.3.1 m:GetInboxRulesRequestType Complex Type ................... 27
          3.1.4.1.3.2 m:GetInboxRulesResponseType Complex Type ............. 27
          3.1.4.1.3.3 t:ArrayOfRulesType Complex Type .................................. 28
        3.1.4.1.4 Simple Types .............................................................................. 29
        3.1.4.1.5 Attributes ................................................................................... 29
        3.1.4.1.6 Groups ........................................................................................ 29
        3.1.4.1.7 Attribute Groups ....................................................................... 29
      3.1.4.2 UpdateInboxRules ............................................................................ 29
        3.1.4.2.1 Messages ................................................................................... 30
          3.1.4.2.1.1 tns:UpdateInboxRulesSoapIn Message ................................ 30
1 Introduction

The Inbox Rules Web Service Protocol is used to get Inbox rules and update Inbox rules for messages on the server.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

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

**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 Secure (HTTPS)**: An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [SSL3] and [RFC5246].

**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-OXWSSCORE] 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".


1.2.2 Informative References


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-OXDSCLI], 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.

![Diagram showing the relationship of Inbox Rules Web Service Protocol to other protocols]

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-OXDSCLI], 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 2.2.3.9, and the version of the server responding to the request is identified using the `t:ServerVersionInfo` element, as described in [MS-OXWSCDATA] section 2.2.3.10.

• **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.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>soap</td>
<td><a href="http://schemas.xmlsoap.org/wsd/soap/">http://schemas.xmlsoap.org/wsd/soap/</a></td>
<td>[SOAP1.1]</td>
</tr>
<tr>
<td>tns</td>
<td><a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a></td>
<td></td>
</tr>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>[XMLSCHEMA1]</td>
</tr>
<tr>
<td>m</td>
<td><a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a></td>
<td></td>
</tr>
<tr>
<td>wsdl</td>
<td><a href="http://schemas.xmlsoap.org/wsd/">http://schemas.xmlsoap.org/wsd/</a></td>
<td>[WSDL]</td>
</tr>
<tr>
<td>t</td>
<td><a href="http://schemas.microsoft.com/exchange/services/2006/types">http://schemas.microsoft.com/exchange/services/2006/types</a></td>
<td></td>
</tr>
</tbody>
</table>

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.

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

2.2.4.1 t:RuleActionsType Complex Type

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

```xml
<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"
      minOccurs="0"
      maxOccurs="1"
  </xs:sequence>
</xs:complexType>
```
The following table lists the child elements of the **RuleActionsType** complex type.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssignCategories</td>
<td>t:ArrayOfStringsType ([MS-OXWSCDATA] section 2.2.4.13)</td>
<td>The categories that are stamped on email messages.</td>
</tr>
<tr>
<td>CopyToFolder</td>
<td>t:TargetFolderIdType ([MS-OXWSFOLD] section 2.2.4.16)</td>
<td>The ID of the folder that email items will be copied to.</td>
</tr>
<tr>
<td>Delete</td>
<td>xs:boolean ([XMLSCHEMA2])</td>
<td>A value that indicates whether messages are to be moved to the <strong>Deleted Items folder</strong>.</td>
</tr>
<tr>
<td>ForwardAsAttachmentToRecipients</td>
<td>t:ArrayOfEmailAddressesType ([MS-OXWSCDATA] section 2.2.4.7)</td>
<td>The email addresses to which messages are to be forwarded as attachments.</td>
</tr>
<tr>
<td>ForwardToRecipients</td>
<td>t:ArrayOfEmailAddressesType</td>
<td>The email addresses to which messages are to be forwarded.</td>
</tr>
<tr>
<td>MarkImportance</td>
<td>t:ImportanceChoicesType ([MS-OXWSCORE] section 2.2.5.3)</td>
<td>The importance that is to be stamped on messages.</td>
</tr>
<tr>
<td>MarkAsRead</td>
<td>xs:boolean</td>
<td>A value that indicates whether messages are to be marked as read.</td>
</tr>
<tr>
<td>MoveToFolder</td>
<td>t:TargetFolderIdType</td>
<td>The ID of the folder that email items will be moved to.</td>
</tr>
<tr>
<td>PermanentDelete</td>
<td>xs:boolean</td>
<td>A value that indicates whether messages are to be deleted.</td>
</tr>
</tbody>
</table>
Element name | Type | Description
--- | --- | ---
 |  | permanently deleted and not saved to the Deleted Items folder.
RedirectToRecipients | t:ArrayOfEmailAddressesType | The email addresses to which messages are to be redirected.
SendSMAalertToRecipients | t:ArrayOfEmailAddressesType | The mobile phone numbers to which a Short Message Service (SMS) alert is to be sent.
ServerReplyWithMessage | t:ItemIdType ([MS-OXWSCORE] section 2.2.4.25) | 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.

```xml
<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.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StartDateTime</td>
<td>xs:dateTime ([XMSCHHEMA2])</td>
<td>Specifies the rule (2) time period and indicates that the rule (2) condition is met after this value.</td>
</tr>
<tr>
<td>EndDateTime</td>
<td>xs:dateTime</td>
<td>Specifies the rule (2) time period and indicates that the rule (2) condition is met before this value.</td>
</tr>
</tbody>
</table>

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.

```xml
<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.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MinimumSize</td>
<td>xs:int</td>
<td>The minimum size of a message for the conditions or exception to apply.</td>
</tr>
<tr>
<td></td>
<td><img src="#" alt="XMLSCHEMA2" /></td>
<td></td>
</tr>
<tr>
<td>MaximumSize</td>
<td>xs:int</td>
<td>The maximum size of a message for the conditions or exception to apply.</td>
</tr>
</tbody>
</table>

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).

```xml
<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:sequence>
</xs:complexType>
```
<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:ArrayOfEmailAddress" minOccurs="0" maxOccurs="1" />
<xs:element name="SentToMe" type="xs:boolean" minOccurs="0" maxOccurs="1" />
The following table lists the child elements of the `RulePredicatesType` complex type.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Categories</td>
<td><code>t:ArrayOfStringsType</code> ([MS-OXWSCDATA] section 2.2.4.13)</td>
<td>The categories that an incoming message is labeled with in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>ContainsBodyStrings</td>
<td><code>t:ArrayOfStringsType</code></td>
<td>The strings that appear in the body of incoming messages in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>ContainsHeaderStrings</td>
<td><code>t:ArrayOfStringsType</code></td>
<td>The strings that appear in the headers of incoming messages in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>ContainsRecipientStrings</td>
<td><code>t:ArrayOfStringsType</code></td>
<td>The strings that appear in either the <code>ToRecipients</code> or <code>CcRecipients</code> properties of incoming messages in order for the condition or exception to apply. The <code>ToRecipients</code> property specifies a collection of recipients of an email. The <code>CcRecipients</code> property specifies a collection of recipients that receive a carbon copy (Cc) of an email.</td>
</tr>
<tr>
<td>ContainsSenderStrings</td>
<td><code>t:ArrayOfStringsType</code></td>
<td>The strings that appear in the <code>From</code> property of incoming messages in order for the condition or exception to apply. The <code>From</code> property specifies the addressee from whom the message was sent.</td>
</tr>
<tr>
<td>ContainsSubjectOrBodyStrings</td>
<td><code>t:ArrayOfStringsType</code></td>
<td>The strings that appear in either the body or the subject of incoming messages in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>Element name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ContainsSubjectStrings</td>
<td>t:ArrayOfStringsType</td>
<td>The strings that appear in the subject of incoming messages in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>FlaggedForAction</td>
<td>t:FlaggedForActionType</td>
<td>The flag for action value that appears on incoming messages in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>FromAddresses</td>
<td>t:ArrayOfEmailAddressType</td>
<td>The email addresses of the senders of incoming messages that appear in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>FromConnectedAccounts</td>
<td>t:ArrayOfStringsType</td>
<td>The email account names from which incoming messages have to have been aggregated in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>HasAttachments</td>
<td>xs:boolean</td>
<td>A value that indicates whether incoming messages have to have attachments in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>Importance</td>
<td>t:ImportanceChoicesType</td>
<td>The importance that is stamped on incoming messages in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>IsApprovalRequest</td>
<td>xs:boolean</td>
<td>A value that indicates whether incoming messages have to be approval requests in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>IsAutomaticForward</td>
<td>xs:boolean</td>
<td>A value that indicates whether incoming messages have to be automatic forwards in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>IsAutomaticReply</td>
<td>xs:boolean</td>
<td>A value that indicates whether incoming messages have to be automatic replies in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>IsEncrypted</td>
<td>xs:boolean</td>
<td>A value that indicates whether incoming messages have to be S/MIME encrypted in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>IsMeetingRequest</td>
<td>xs:boolean</td>
<td>A value that indicates whether incoming messages have to be meeting requests in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>IsMeetingResponse</td>
<td>xs:boolean</td>
<td>A value that indicates whether incoming messages have to be meeting responses in order for the condition or exception to apply.</td>
</tr>
</tbody>
</table>
| 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.
<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsPermissionControlled</td>
<td>xs:boolean</td>
<td>A value that indicates whether incoming messages have to be permission controlled (RMS protected) in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>IsReadReceipt</td>
<td>xs:boolean</td>
<td>A value that indicates whether incoming messages have to be read receipts in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>IsSigned</td>
<td>xs:boolean</td>
<td>A value that indicates whether incoming messages have to be S/MIME signed in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>IsVoicemail</td>
<td>xs:boolean</td>
<td>A value that indicates whether incoming messages have to be voice mails in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>ItemClasses</td>
<td>t:ArrayOfStringsType</td>
<td>The item classes that have to be stamped on incoming messages in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>MessageClassifications</td>
<td>t:ArrayOfStringsType</td>
<td>The message classifications that have to be stamped on incoming messages in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>NotSentToMe</td>
<td>xs:boolean</td>
<td>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.</td>
</tr>
<tr>
<td>SentCcMe</td>
<td>xs:boolean</td>
<td>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.</td>
</tr>
<tr>
<td>SentOnlyToMe</td>
<td>xs:boolean</td>
<td>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 to apply.</td>
</tr>
<tr>
<td>SentToAddresses</td>
<td>t:ArrayOfEmailAddressesType</td>
<td>The email addresses that incoming messages have to have been sent to in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>SentToMe</td>
<td>xs:boolean</td>
<td>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.</td>
</tr>
<tr>
<td>Element name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SentToOrCcMe</td>
<td>xs:boolean</td>
<td>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.</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>t:SensitivityChoicesType ([MS-OXWSCDATA] section 2.2.5.25)</td>
<td>The sensitivity that has to be stamped on incoming messages in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>WithinDateRange</td>
<td>t:RulePredicateDateRangeType (section 2.2.4.2)</td>
<td>The date range within which incoming messages have to have been received in order for the condition or exception to apply.</td>
</tr>
<tr>
<td>WithinSizeRange</td>
<td>t:RulePredicateSizeRangeType (section 2.2.4.3)</td>
<td>The minimum and maximum sizes that incoming messages have to have in order for the condition or exception to apply.</td>
</tr>
</tbody>
</table>

2.2.4.5 t:RuleType Complex Type

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

```xml
<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" maxOccurs="1" minOccurs="1"/>
    <xs:element name="IsEnabled" type="xs:boolean" maxOccurs="1" 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"/>
  </xs:sequence>
</xs:complexType>
```
The following table lists the child elements of the `RuleType` complex type.

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

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.

<table>
<thead>
<tr>
<th>Simple type name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlaggedForActionType</td>
<td>Defines the follow-up actions that can be stamped on a message.</td>
</tr>
</tbody>
</table>
2.2.5.1 t:FlaggedForActionType Simple Type

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

```xml
<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.

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

### 2.2.6 Attributes

This specification does not define any common [XML schema](https://www.w3.org/XML/namespace) attribute definitions.

### 2.2.7 Groups

This specification does not define any common [XML schema](https://www.w3.org/XML/namespace) group definitions.

### 2.2.8 Attribute Groups

This specification does not define any common [XML schema](https://www.w3.org/XML/namespace) 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.

<table>
<thead>
<tr>
<th>Operation name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInboxRules</td>
<td>Retrieves a collection of Inbox rules (2) in the identified user's mailbox.</td>
</tr>
<tr>
<td>UpdateInboxRules</td>
<td>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).</td>
</tr>
</tbody>
</table>

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.

```xml
<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.

```xml
<wsdl:operation name="GetInboxRules">
  <soap:operation
  <wsdl:input>
    <soap:header message="tns:GetInboxRulesSoapIn" part="Impersonation" use="literal" />
  </wsdl:input>
```
The protocol client sends a GetInboxRulesSoapIn request WSDL message and the protocol server responds with a GetInboxRulesSoapOut response WSDL message.


### 3.1.4.1.1 Messages

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

<table>
<thead>
<tr>
<th>Message name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInboxRulesSoapIn</td>
<td>Specifies the SOAP message that gets the Inbox rules (2).</td>
</tr>
<tr>
<td>GetInboxRulesSoapOut</td>
<td>Specifies the SOAP message that is returned by the server in response.</td>
</tr>
</tbody>
</table>

#### 3.1.4.1.1.1 tns:GetInboxRulesSoapIn Message

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

```xml
<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 five parts of the GetInboxRulesSoapIn message are described in the following table.

<table>
<thead>
<tr>
<th>Part name</th>
<th>Element/type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>tns:GetInboxRules (section 3.1.4.1.2.1)</td>
<td>Specifies the SOAP body of the request to get the Inbox rules (2).</td>
</tr>
<tr>
<td>Impersonation</td>
<td>t:ExchangeImpersonation ([MS-OXWSCDATA] section 2.2.3.3)</td>
<td>Specifies a SOAP header that identifies the user whom the client application is impersonating.</td>
</tr>
<tr>
<td>MailboxCulture</td>
<td>t:MailboxCulture ([MS-OXWSCDATA] section 2.2.3.6)</td>
<td>Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are</td>
</tr>
</tbody>
</table>
### 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).

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


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

<table>
<thead>
<tr>
<th>Part name</th>
<th>Element/type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInboxRulesResult</td>
<td>tns:GetInboxRulesResponse (section 3.1.4.1.2.2)</td>
<td>Specifies the SOAP body of the response to a <strong>GetInboxRules</strong> operation request.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>t:ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.10)</td>
<td>Specifies a SOAP header that identifies the server version for the response.</td>
</tr>
</tbody>
</table>

### 3.1.4.1.2 Elements

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

<table>
<thead>
<tr>
<th>Element name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInboxRules</td>
<td>Specifies a request to get the Inbox rules (2) in a mailbox in the server message store.</td>
</tr>
<tr>
<td>GetInboxRulesResponse</td>
<td>Specifies the response body content from a request to get Inbox rules (2).</td>
</tr>
</tbody>
</table>

### 3.1.4.1.2.1  

**GetInboxRules Element**

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

```xml
<xs:element name="GetInboxRules" type="m:GetInboxRulesRequestType" />
```
3.1.4.1.2.2 GetInboxRulesResponse Element


```xml
<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.

<table>
<thead>
<tr>
<th>Complex type name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetInboxRulesRequestType</td>
<td>Specifies a request message for the GetInboxRules operation.</td>
</tr>
<tr>
<td>GetInboxRulesResponseType</td>
<td>Specifies a response message for the GetInboxRules operation.</td>
</tr>
<tr>
<td>ArrayOfRulesType</td>
<td>Specifies an array of RuleType complex type objects.</td>
</tr>
</tbody>
</table>

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.

```xml
<xs:complexType name="GetInboxRulesRequestType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:sequence>
        <xs:element name="MailboxSmtpAddress"
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.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MailboxSmtpAddress</td>
<td>xs:string ([XMLSCHEMA2])</td>
<td>Represents the SMTP address of the user whose Inbox rules (2) are to be retrieved.</td>
</tr>
</tbody>
</table>
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 2.2.4.67.

```xml
<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.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutlookRuleBlobExists</td>
<td>xs:boolean ([XMLSCHEMA2])</td>
<td>Specifies whether a binary large object (BLOB) that contains all rules (2) exists in the user's mailbox. A text value of &quot;true&quot; indicates that the BLOB exists. This element MUST be present.</td>
</tr>
<tr>
<td>InboxRules</td>
<td>t:ArrayOfRulesType (section 3.1.4.1.3.3)</td>
<td>Represents the response message and includes an array of the rules (2) in the user's mailbox.</td>
</tr>
</tbody>
</table>

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.

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

The following table lists the child element of the ArrayOfRulesType complex type.
<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule</td>
<td>t:RuleType (section 2.2.4.5)</td>
<td>The rule (2) in the user's mailbox.</td>
</tr>
</tbody>
</table>

### 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.

```xml
<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.

```xml
<wsdl:operation name="UpdateInboxRules">
  <soap:operation
  <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>
```
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.

<table>
<thead>
<tr>
<th>Message name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateInboxRulesSoapIn</td>
<td>Specifies the <strong>SOAP message</strong> that updates the Inbox rules (2).</td>
</tr>
<tr>
<td>UpdateInboxRulesSoapOut</td>
<td>Specifies the SOAP message that is returned by the server in response.</td>
</tr>
</tbody>
</table>

#### 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).

```xml
<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.

<table>
<thead>
<tr>
<th>Part name</th>
<th>Element/type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td><strong>tns:UpdateInboxRules</strong> (<a href="#">section 3.1.4.2.2.1</a>)</td>
<td>Specifies the <strong>SOAP body</strong> of the request to update Inbox rules (2).</td>
</tr>
<tr>
<td>Impersonation</td>
<td><strong>t:ExchangeImpersonation</strong> (<a href="#">MS-OXWSDATA</a> section 2.2.3.3)</td>
<td>Specifies a <strong>SOAP header</strong> that identifies the user whom the client application is impersonating.</td>
</tr>
<tr>
<td>MailboxCulture</td>
<td><strong>t:MailboxCulture</strong> (<a href="#">MS-OXWSDATA</a> section 2.2.3.6)</td>
<td>Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].</td>
</tr>
<tr>
<td>RequestVersion</td>
<td><strong>t:RequestServerVersion</strong> (<a href="#">MS-OXWSDATA</a> section 2.2.3.9)</td>
<td>Specifies a SOAP header that identifies the schema version for the <strong>UpdateInboxRules</strong> operation request.</td>
</tr>
<tr>
<td>TimeZoneContext</td>
<td><strong>t:TimeZoneContext</strong> (<a href="#">MS-OXWSGTZ</a> section 2.2.3.4)</td>
<td>Specifies a time zone definition and enables the association of SOAP attributes with the definition.</td>
</tr>
</tbody>
</table>
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).

```xml
<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.

<table>
<thead>
<tr>
<th>Part name</th>
<th>Element/type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateInboxRulesResult</td>
<td>tns:UpdateInboxRulesResponse (section 3.1.4.2.2.2)</td>
<td>Specifies the response to an UpdateInboxRules operation request.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>t:ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.10)</td>
<td>Specifies the server version for the response.</td>
</tr>
</tbody>
</table>

3.1.4.2.2 Elements

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

<table>
<thead>
<tr>
<th>Element name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateInboxRules</td>
<td>Specifies a request to update the Inbox rules (2) in a mailbox in the server message store.</td>
</tr>
<tr>
<td>UpdateInboxRulesResponse</td>
<td>Specifies the response body content from a request to update the Inbox rules (2).</td>
</tr>
</tbody>
</table>

3.1.4.2.2.1 UpdateInboxRules Element

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

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

3.1.4.2.2.2 UpdateInboxRulesResponse Element

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

```xml
<xs:element name="UpdateInboxRulesResponse"/>
3.1.4.2.3 Complex Types

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

<table>
<thead>
<tr>
<th>Complex type name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateInboxRulesRequestType</td>
<td>Specifies the request message for the UpdateInboxRules operation.</td>
</tr>
<tr>
<td>UpdateInboxRulesResponseType</td>
<td>Specifies the response message for the UpdateInboxRules operation.</td>
</tr>
<tr>
<td>ArrayOfRuleOperationErrorsType</td>
<td>Specifies an array of rule operations that can be performed on a rule.</td>
</tr>
<tr>
<td>RuleOperationType</td>
<td>Specifies the base type for rule operations and represents the rule operation to be performed.</td>
</tr>
<tr>
<td>CreateRuleOperationType</td>
<td>Specifies an operation to create a new rule.</td>
</tr>
<tr>
<td>SetRuleOperationType</td>
<td>Specifies an operation to update an existing rule.</td>
</tr>
<tr>
<td>DeleteRuleOperationType</td>
<td>Specifies an operation to delete an existing rule.</td>
</tr>
<tr>
<td>ArrayOfRuleOperationErrorsType</td>
<td>Specifies an array of rule validation errors on each rule field that has an error.</td>
</tr>
<tr>
<td>RuleOperationErrorType</td>
<td>Specifies a rule operation error.</td>
</tr>
<tr>
<td>ArrayOfRuleValidationErrorsType</td>
<td>Specifies an array of rule validation errors.</td>
</tr>
<tr>
<td>RuleValidation&gt;ErrorType</td>
<td>Specifies a single validation error on a particular rule property value, predicate property value, or action property value.</td>
</tr>
</tbody>
</table>

3.1.4.2.3.1 m:UpdateInboxRulesRequestType Complex Type

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

```xml
<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="m:ArrayOfRuleOperationsType" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```
The following table lists the child elements of the **UpdateInboxRulesRequestType** complex type.

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

### 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 2.2.4.67.

```xml
<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.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleOperationErrors</td>
<td>t:ArrayOfRuleOperationErrorsType (section 3.1.4.2.3.8)</td>
<td>Specifies a <strong>RuleOperationErrorType</strong> complex type (section 3.1.4.2.3.9) collection, which contains the validation errors on each rule (2) filed that has an error.</td>
</tr>
</tbody>
</table>
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).

```xml
<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.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateRuleOperation</td>
<td>t:CreateRuleOperationType (section 3.1.4.2.3.5)</td>
<td>Specifies an operation to create an Inbox rule (2).</td>
</tr>
<tr>
<td>SetRuleOperation</td>
<td>t:SetRuleOperationType (section 3.1.4.2.3.6)</td>
<td>Specifies an operation to update an Inbox rule (2).</td>
</tr>
<tr>
<td>DeleteRuleOperation</td>
<td>t:DeleteRuleOperationType (section 3.1.4.2.3.7)</td>
<td>Specifies an operation to delete an Inbox rule (2).</td>
</tr>
</tbody>
</table>

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.

```xml
<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.

```xml
<xs:complexType name="CreateRuleOperationType">
</xs:complexType>
```
The following table lists the child element of the `CreateRuleOperationType` complex type.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule</td>
<td><code>t:RuleType</code></td>
<td>(section 2.2.4.5) Represents the rule (2) to create.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule</td>
<td><code>t:RuleType</code></td>
<td>(section 2.2.4.5) Represents the rule (2) to be updated.</td>
</tr>
</tbody>
</table>

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  \textit{t:DeleteRuleOperationType} Complex Type

The \textit{DeleteRuleOperationType} complex type represents an operation to delete an existing Inbox rule (2). The \textit{DeleteRuleOperationType} complex type extends the \textit{RuleOperationType} complex type, as specified in section 3.1.4.2.3.4.

\begin{verbatim}
<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>
\end{verbatim}

The following table lists the child element of the \textit{DeleteRuleOperationType} complex type.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleId</td>
<td>\textit{xs:string} (\text{(XMLSCHEMA2)})</td>
<td>Specifies the rule (2) identifier of the Inbox rule (2) to be deleted.</td>
</tr>
</tbody>
</table>

3.1.4.2.3.8  \textit{t:ArrayOfRuleOperationErrorsType} Complex Type

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

\begin{verbatim}
<xs:complexType name="ArrayOfRuleOperationErrorsType">
  <xs:sequence>
    <xs:element name="RuleOperationError" type="t:RuleOperationErrorType" maxOccurs="unbounded" minOccurs="1"/>
  </xs:sequence>
</xs:complexType>
\end{verbatim}

The following table lists the child element of the \textit{ArrayOfRuleOperationErrorsType} complex type.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleOperationError</td>
<td>\textit{t:RuleOperationErrorType} (\text{(section 3.1.4.2.3.9)})</td>
<td>Specifies a rule (2) operation error.</td>
</tr>
</tbody>
</table>

3.1.4.2.3.9  \textit{t:RuleOperationErrorType} Complex Type
The **RuleOperationErrorType** complex type specifies a rule (2) operation error.

```xml
<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.

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

3.1.4.2.3.10 t:ArrayOfRuleValidationErrorsType Complex Type

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

```xml
<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.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>t:RuleValidationErrorType (section 3.1.4.2.3.11)</td>
<td>Specifies a single validation error on a particular rule (2) property value, predicate property value, or action property value.</td>
</tr>
</tbody>
</table>

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.

```xml
<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.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FieldURI</td>
<td>t:RuleFieldURIType</td>
<td>Specifies the <strong>URI</strong> to the rule (2) field that caused the validation error.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>t:RuleValidationErrorCodeType</td>
<td>Specifies the error code that was triggered on the field.</td>
</tr>
<tr>
<td>ErrorMessage</td>
<td>xs:string (IXMLSCHEMA2)</td>
<td>Specifies the reason for the validation error.</td>
</tr>
<tr>
<td>FieldValue</td>
<td>xs:string</td>
<td>Specifies the value of the field that caused the validation error. This element is optional.</td>
</tr>
</tbody>
</table>

### 3.1.4.2.4 Simple Types

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

<table>
<thead>
<tr>
<th>Simple type name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleFieldURIType</td>
<td>Defines the <strong>rule (2)</strong> fields that could cause a validation error.</td>
</tr>
<tr>
<td>RuleValidationErrorCodeType</td>
<td>Defines what failed validation for each rule (2) predicate or <strong>action</strong>.</td>
</tr>
</tbody>
</table>

#### 3.1.4.2.4.1  t:RuleFieldURIType Simple Type
The **RuleFieldURIType** simple type defines all possible rule fields that could trigger a validation error.

```xml
<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:restriction>
</xs:simpleType>
```
</xs:enumeration>
<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"/>
The following table lists the values that are defined by the `RuleFieldURIType` simple type.

<table>
<thead>
<tr>
<th>Value name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleId</td>
<td>The rule (2) ID.</td>
</tr>
<tr>
<td>DisplayName</td>
<td>The rule (2) display name.</td>
</tr>
<tr>
<td>Priority</td>
<td>The rule (2) priority.</td>
</tr>
<tr>
<td>IsNotSupported</td>
<td>The rule (2) is not supported.</td>
</tr>
<tr>
<td>Actions</td>
<td>The rule (2) actions.</td>
</tr>
<tr>
<td>Condition:Categories</td>
<td>The categories condition.</td>
</tr>
<tr>
<td>Condition:ContainsBodyStrings</td>
<td>The body contains strings condition.</td>
</tr>
<tr>
<td>Condition:ContainsHeaderStrings</td>
<td>The header contains strings condition.</td>
</tr>
<tr>
<td>Condition:ContainsRecipientStrings</td>
<td>The recipient contains strings condition.</td>
</tr>
<tr>
<td>Condition:ContainsSenderStrings</td>
<td>The sender contains strings condition.</td>
</tr>
<tr>
<td>Condition:ContainsSubjectOrBodyStrings</td>
<td>The subject or body contains strings condition.</td>
</tr>
<tr>
<td>Condition:ContainsSubjectStrings</td>
<td>The subject contains strings condition.</td>
</tr>
<tr>
<td>Condition:FlaggedForAction</td>
<td>The flagged for action condition.</td>
</tr>
<tr>
<td>Condition:FromAddresses</td>
<td>The from addresses condition.</td>
</tr>
<tr>
<td>Condition:FromConnectedAccounts</td>
<td>The from connected accounts condition.</td>
</tr>
<tr>
<td>Condition:HasAttachments</td>
<td>The has attachments condition.</td>
</tr>
<tr>
<td>Condition:Importance</td>
<td>The importance condition property of a rule's (2) set of conditions.</td>
</tr>
<tr>
<td>Condition:IsApprovalRequest</td>
<td>The approval request condition exists.</td>
</tr>
<tr>
<td>Condition:IsAutomaticForward</td>
<td>The automatic forward condition exists.</td>
</tr>
<tr>
<td>Condition:IsAutomaticReply</td>
<td>The automatic reply condition exists.</td>
</tr>
<tr>
<td>Condition:IsEncrypted</td>
<td>The encrypted condition exists.</td>
</tr>
<tr>
<td>Condition:IsMeetingRequest</td>
<td>The meeting request condition exists.</td>
</tr>
<tr>
<td>Condition:IsMeetingResponse</td>
<td>The meeting response condition exists.</td>
</tr>
<tr>
<td>Condition:IsNDR</td>
<td>The non-delivery report (NDR) condition exists.</td>
</tr>
<tr>
<td>Condition:IsPermissionControlled</td>
<td>The permission-controlled condition exists.</td>
</tr>
<tr>
<td>Condition:IsReadReceipt</td>
<td>The read condition exists.</td>
</tr>
<tr>
<td>Condition:IsSigned</td>
<td>The signed condition exists.</td>
</tr>
<tr>
<td>Value name</td>
<td>Meaning</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Condition: IsVoicemail</td>
<td>The voice mail condition exists.</td>
</tr>
<tr>
<td>Condition: ItemClasses</td>
<td>The item class condition.</td>
</tr>
<tr>
<td>Condition: MessageClassifications</td>
<td>The message classifications condition.</td>
</tr>
<tr>
<td>Condition: NotSentToMe</td>
<td>The not sent to me condition.</td>
</tr>
<tr>
<td>Condition: SentCcMe</td>
<td>The sent Cc me condition.</td>
</tr>
<tr>
<td>Condition: SentOnlyToMe</td>
<td>The sent only to me condition.</td>
</tr>
<tr>
<td>Condition: SentToAddresses</td>
<td>The sent to addresses condition.</td>
</tr>
<tr>
<td>Condition: SentToMe</td>
<td>The sent to me condition.</td>
</tr>
<tr>
<td>Condition: SentToOrCcMe</td>
<td>The sent to or Cc me condition.</td>
</tr>
<tr>
<td>Condition: Sensitivity</td>
<td>The sensitivity condition.</td>
</tr>
<tr>
<td>Condition: WithinDateRange</td>
<td>The within date range condition.</td>
</tr>
<tr>
<td>Condition: WithinSizeRange</td>
<td>The within size range condition.</td>
</tr>
<tr>
<td>Exception: Categories</td>
<td>The categories exception.</td>
</tr>
<tr>
<td>Exception: ContainsBodyStrings</td>
<td>The body contains strings exception.</td>
</tr>
<tr>
<td>Exception: ContainsHeaderStrings</td>
<td>The header contains strings exception.</td>
</tr>
<tr>
<td>Exception: ContainsRecipientStrings</td>
<td>The recipient contains strings exception.</td>
</tr>
<tr>
<td>Exception: ContainsSenderStrings</td>
<td>The sender contains strings exception.</td>
</tr>
<tr>
<td>Exception: ContainsSubjectOrBodyStrings</td>
<td>The subject or body contains strings exception.</td>
</tr>
<tr>
<td>Exception: ContainsSubjectStrings</td>
<td>The subject contains strings exception.</td>
</tr>
<tr>
<td>Exception: FlaggedForAction</td>
<td>The flagged for action exception.</td>
</tr>
<tr>
<td>Exception: FromAddresses</td>
<td>The from addresses exception.</td>
</tr>
<tr>
<td>Exception: FromConnectedAccounts</td>
<td>The from connected accounts exception.</td>
</tr>
<tr>
<td>Exception: HasAttachments</td>
<td>The has attachments exception.</td>
</tr>
<tr>
<td>Exception: Importance</td>
<td>The importance exception.</td>
</tr>
<tr>
<td>Exception: IsApprovalRequest</td>
<td>The approval request exception exists.</td>
</tr>
<tr>
<td>Exception: IsAutomaticForward</td>
<td>The automatic forward exception exists.</td>
</tr>
<tr>
<td>Exception: IsAutomaticReply</td>
<td>The automatic reply exception exists.</td>
</tr>
<tr>
<td>Exception: IsEncrypted</td>
<td>The encrypted exception exists.</td>
</tr>
<tr>
<td>Exception: IsMeetingRequest</td>
<td>The meeting request exception exists.</td>
</tr>
<tr>
<td>Exception: IsMeetingResponse</td>
<td>The meeting response exception exists.</td>
</tr>
<tr>
<td>Exception: IsNDR</td>
<td>The NDR exception exists.</td>
</tr>
<tr>
<td>Exception: IsPermissionControlled</td>
<td>The permission controlled exception exists.</td>
</tr>
<tr>
<td>Value name</td>
<td>Meaning</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------</td>
</tr>
<tr>
<td>Exception:IsReadReceipt</td>
<td>The read exception exists.</td>
</tr>
<tr>
<td>Exception:IsSigned</td>
<td>The signed exception exists.</td>
</tr>
<tr>
<td>Exception:IsVoicemail</td>
<td>The voice mail exception exists.</td>
</tr>
<tr>
<td>Exception:ItemClasses</td>
<td>The item class exception.</td>
</tr>
<tr>
<td>Exception:MessageClassifications</td>
<td>The message classifications exception.</td>
</tr>
<tr>
<td>Exception:NotSentToMe</td>
<td>The not sent to me exception.</td>
</tr>
<tr>
<td>Exception:SentCcMe</td>
<td>The sent to me exception.</td>
</tr>
<tr>
<td>Exception:SentOnlyToMe</td>
<td>The sent only to me exception.</td>
</tr>
<tr>
<td>Exception:SentToAddresses</td>
<td>The sent to addresses exception.</td>
</tr>
<tr>
<td>Exception:SentToMe</td>
<td>The sent to me exception.</td>
</tr>
<tr>
<td>Exception:SentToOrCcMe</td>
<td>The sent to or Cc me exception.</td>
</tr>
<tr>
<td>Exception:Sensitivity</td>
<td>The sensitivity exception.</td>
</tr>
<tr>
<td>Exception:WithinDateRange</td>
<td>The within date range exception.</td>
</tr>
<tr>
<td>Exception:WithinSizeRange</td>
<td>The within size range exception.</td>
</tr>
<tr>
<td>Action:AssignCategories</td>
<td>The assign categories action.</td>
</tr>
<tr>
<td>Action:CopyToFolder</td>
<td>The copy to folder action.</td>
</tr>
<tr>
<td>Action:Delete</td>
<td>The delete action.</td>
</tr>
<tr>
<td>Action:ForwardAsAttachmentToRecipients</td>
<td>The forward as attachment to recipients action.</td>
</tr>
<tr>
<td>Action:ForwardToRecipients</td>
<td>The forward to recipients action.</td>
</tr>
<tr>
<td>Action:MarkImportance</td>
<td>The mark importance action.</td>
</tr>
<tr>
<td>Action:MarkAsRead</td>
<td>The mark as read action.</td>
</tr>
<tr>
<td>Action:MoveToFolder</td>
<td>The move to folder action.</td>
</tr>
<tr>
<td>Action:PermanentDelete</td>
<td>The permanent delete action.</td>
</tr>
<tr>
<td>Action:RedirectToRecipients</td>
<td>The redirect to recipients action.</td>
</tr>
<tr>
<td>Action:SendSMSAlertToRecipients</td>
<td>The send SMS alert to recipients action.</td>
</tr>
<tr>
<td>Action:ServerReplyWithMessage</td>
<td>The server reply with message action.</td>
</tr>
<tr>
<td>Action:StopProcessingRules</td>
<td>The stop processing rules action.</td>
</tr>
<tr>
<td>IsEnabled</td>
<td>Indicates whether the rule (2) is enabled.</td>
</tr>
<tr>
<td>IsInError</td>
<td>Indicates whether the rule (2) is in error.</td>
</tr>
<tr>
<td>Conditions</td>
<td>Contains all conditions of the rule (2).</td>
</tr>
<tr>
<td>Exceptions</td>
<td>Contains all the exceptions for the rule (2).</td>
</tr>
</tbody>
</table>
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.

```xml
<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:restriction>
</xs:simpleType>
```
The following table lists the values that are defined by the `RuleValidationErrorCodeType` simple type.

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

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

```xml
<?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>CreateRuleOperation</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
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    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>
```

[MS-OXWSRULES] - v20220215
Inbox Rules Web Service Protocol
Copyright © 2022 Microsoft Corporation
Release: February 15, 2022
The server constructs the response XML and sends it to the client. The RuleValidationErrorCode 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/>
  <s:Body>
    <UpdateInboxRulesResponse ResponseClass="Error" xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <MessageText>A validation error occurred while executing the rule operation.</MessageText>
      <ResponseCode>ErrorInboxRulesValidation</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>

The server constructs the response XML and sends it to the client. The RuleValidationErrorCode 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.
<s:Body>
</s:Body>
</s:Envelope>
5 Security

5.1 Security Considerations for Implementers
None.

5.2 Index of Security Parameters
None.
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.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-OXWSRULES.wsdl</td>
<td>Contains the WSDL for the implementation of this protocol.</td>
<td>6</td>
</tr>
<tr>
<td>MS-OXWSRULES-messages.xsd</td>
<td>Contains the XML schema message definitions that are used in this protocol.</td>
<td>7.1</td>
</tr>
<tr>
<td>MS-OXWSRULES-types.xsd</td>
<td>Contains the XML schema type definitions that are used in this protocol.</td>
<td>7.2</td>
</tr>
</tbody>
</table>

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
<?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="http://schemas.microsoft.com/exchange/services/2006/types">
<wsdl:types>
 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
schemaLocation="MS-OXWSRULES-messages.xsd"/>
</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>
<!-- Inbox Rules -->
<wsdl:operation name="GetInboxRules">
 <wsdl:input message="tns:GetInboxRulesSoapIn" />
</wsdl:operation>

```
<wsdl:operation name="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" />
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="GetInboxRulesResult" use="literal" />
        <soap:header message="tns:GetInboxRulesSoapOut" part="ServerVersion" use="literal" />
    </wsdl:output>
</wsdl:operation>

<wsdl:operation name="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:output>
        <soap:body parts="UpdateInboxRulesResult" use="literal" />
        <soap:header message="tns:UpdateInboxRulesSoapOut" part="ServerVersion" use="literal" />
    </wsdl:output>
</wsdl:operation>
7  Appendix B: Full XML Schema

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

<table>
<thead>
<tr>
<th>Schema name</th>
<th>Prefix</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messages schema</td>
<td>m:</td>
<td>7.1</td>
</tr>
<tr>
<td>Types schema</td>
<td>t:</td>
<td>7.2</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>File name</th>
<th>Defining specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-OXWSCDATA -messages.xsd</td>
<td>[MS-OXWSCDATA] section 7.1</td>
</tr>
</tbody>
</table>

```xml
<?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="MailboxSmtpAddress" 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>
```
61 / 70

[MS-OXWSRULES] - v20220215
Inbox Rules Web Service Protocol
Copyright © 2022 Microsoft Corporation
Release: February 15, 2022

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.

<table>
<thead>
<tr>
<th>File name</th>
<th>Defining specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-OXWSCDATA-types.xsd</td>
<td>[MS-OXWSCDATA] section 7.2</td>
</tr>
<tr>
<td>MS-OXWSCORE-types.xsd</td>
<td>[MS-OXWSCORE] section 7.2</td>
</tr>
</tbody>
</table>

<?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: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:sequence>
    </xs:complexType>
<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="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="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="PermanentlyDelete" type="xs:boolean" minOccurs="0"/>
<xs:element name="RedirectToRecipients" type="t:ArrayOfEmailAddressesType" minOccurs="0"/>
<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: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: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:restriction>
</xs:simpleType>
<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: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:MarkAsImportance"/>
<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: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: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>
</xs:schema>
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 updates to those products.

- Microsoft Exchange Server 2010 Service Pack 1 (SP1)
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Exchange Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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.
9 Change Tracking

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