**[MS-OXWSCEXT]:**

Client Extension 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.
- **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>7/16/2012</td>
<td>0.1</td>
<td>New</td>
<td>Released new document.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>1.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>1.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>7/26/2013</td>
<td>1.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>1.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>1.1</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>1.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>1.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>1.3</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>3/16/2015</td>
<td>2.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>5/26/2015</td>
<td>2.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/14/2015</td>
<td>3.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>6/13/2016</td>
<td>4.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>4.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/19/2017</td>
<td>4.0</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>5.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/1/2018</td>
<td>6.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>3/19/2019</td>
<td>6.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>4/22/2021</td>
<td>7.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>8/17/2021</td>
<td>8.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
</tbody>
</table>
# Table of Contents

1 **Introduction** ........................................................................................................... 6
  1.1 Glossary ................................................................................................................. 6
  1.2 References ............................................................................................................. 7
    1.2.1 Normative References ...................................................................................... 7
    1.2.2 Informative References .................................................................................... 8
  1.3 Overview .................................................................................................................. 8
  1.4 Relationship to Other Protocols ............................................................................ 8
  1.5 Prerequisites/Preconditions ................................................................................... 9
  1.6 Applicability Statement ......................................................................................... 9
  1.7 Versioning and Capability Negotiation ................................................................. 9
  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
    2.2.5 Simple Types ................................................................................................... 11
    2.2.6 Attributes ........................................................................................................ 11
    2.2.7 Groups .............................................................................................................. 11
    2.2.8 Attribute Groups ............................................................................................. 11

3 **Protocol Details** ..................................................................................................... 12
  3.1 ExchangeServicePortType Server Details .......................................................... 12
    3.1.1 Abstract Data Model ......................................................................................... 12
    3.1.2 Timers ............................................................................................................... 12
    3.1.3 Initialization ...................................................................................................... 12
    3.1.4 Message Processing Events and Sequencing Rules ....................................... 12
      3.1.4.1 DisableApp ................................................................................................... 12
        3.1.4.1.1 Messages ............................................................................................... 13
          3.1.4.1.1.1 DisableAppSoapIn ............................................................................ 13
          3.1.4.1.1.2 DisableAppSoapOut ....................................................................... 14
        3.1.4.1.2 Elements ............................................................................................... 14
          3.1.4.1.2.1 DisableApp ....................................................................................... 15
          3.1.4.1.2.2 DisableAppResponse ....................................................................... 15
        3.1.4.1.3 Complex Types ...................................................................................... 15
          3.1.4.1.3.1 DisableAppType .............................................................................. 15
          3.1.4.1.3.2 DisableAppResponseType ............................................................. 16
        3.1.4.1.4 Simple Types ........................................................................................ 16
          3.1.4.1.4.1 DisableReasonType ........................................................................ 16
        3.1.4.1.5 Attributes .............................................................................................. 16
        3.1.4.1.6 Groups ................................................................................................... 17
        3.1.4.1.7 Attribute Groups ................................................................................... 17
      3.1.4.2 GetAppManifests ......................................................................................... 17
        3.1.4.2.1 Messages ............................................................................................... 17
          3.1.4.2.1.1 GetAppManifestsSoapIn ................................................................. 17
          3.1.4.2.1.2 GetAppManifestsSoapOut .............................................................. 18
        3.1.4.2.2 Elements ............................................................................................... 19
        3.1.4.2.3 Complex Types ...................................................................................... 19
          3.1.4.2.3.1 GetAppManifestsType ................................................................. 19
          3.1.4.2.3.2 GetAppManifestsResponse ............................................................. 19
          3.1.4.2.3.3 GetAppManifestsSoapIn ................................................................. 19
          3.1.4.2.3.4 GetAppManifestsSoapOut .............................................................. 19
3.1.4.2.3.2 GetAppManifestsResponseType ........................................ 20
3.1.4.2.3.3ArrayOfAppManifestsType ........................................ 20
3.1.4.2.4 Simple Types .................................................................. 20
3.1.4.2.5 Attributes ................................................................... 20
3.1.4.2.6 Groups ...................................................................... 20
3.1.4.2.7 Attribute Groups ......................................................... 20
3.1.4.3 GetAppMarketplaceUrl ....................................................... 21
3.1.4.3.1 Messages ................................................................... 21
3.1.4.3.1.1 GetAppMarketplaceUrlSoapIn ................................... 21
3.1.4.3.1.2 GetAppMarketplaceUrlSoapOut .................................. 22
3.1.4.3.2 Elements .................................................................. 23
3.1.4.3.2.1 GetAppMarketplaceUrl ........................................... 23
3.1.4.3.2.2 GetAppMarketplaceUrlResponse ................................ 23
3.1.4.3.3 Complex Types ............................................................ 23
3.1.4.3.3.1 GetAppMarketplaceUrlType ..................................... 23
3.1.4.3.3.2 GetAppMarketplaceUrlResponseMessageType ............. 24
3.1.4.3.4 Simple Types ............................................................... 24
3.1.4.3.5 Attributes .................................................................. 24
3.1.4.3.6 Groups ...................................................................... 24
3.1.4.3.7 Attribute Groups ......................................................... 24
3.1.4.4 GetClientAccessToken.......................................................... 24
3.1.4.4.1 Messages ................................................................... 25
3.1.4.4.1.1 GetClientAccessTokenSoapIn ................................... 25
3.1.4.4.1.2 GetClientAccessTokenSoapOut .................................. 26
3.1.4.4.2 Elements .................................................................. 26
3.1.4.4.2.1 GetClientAccessToken ............................................ 27
3.1.4.4.2.2 GetClientAccessTokenResponse ............................... 27
3.1.4.4.3 Complex Types ............................................................ 27
3.1.4.4.3.1 GetClientAccessTokenType ..................................... 27
3.1.4.4.3.2 NonEmptyArrayOfClientAccessTokenRequestsType .... 28
3.1.4.4.3.3 ClientAccessTokenRequestType ................................. 28
3.1.4.4.3.4 GetClientAccessTokenResponseMessageType ............. 28
3.1.4.4.3.5 ClientAccessTokenType ........................................... 29
3.1.4.4.3.6 GetClientAccessTokenResponseType .......................... 29
3.1.4.4.4 Simple Types ............................................................... 29
3.1.4.4.4.1 ClientAccessTokenTokenType .................................. 30
3.1.4.4.5 Attributes .................................................................. 30
3.1.4.4.6 Groups ...................................................................... 30
3.1.4.4.7 Attribute Groups ......................................................... 30
3.1.4.5 InstallApp ........................................................................ 30
3.1.4.5.1 Messages ................................................................... 31
3.1.4.5.1.1 InstallAppSoapIn .................................................... 31
3.1.4.5.1.2 InstallAppSoapOut .................................................. 32
3.1.4.5.2 Elements .................................................................. 32
3.1.4.5.2.1 InstallApp ............................................................... 32
3.1.4.5.2.2 InstallAppResponse .................................................. 33
3.1.4.5.3 Complex Types ............................................................ 33
3.1.4.5.3.1 InstallAppType ........................................................ 33
3.1.4.5.3.2 InstallAppResponseType .......................................... 33
3.1.4.5.4 Simple Types ............................................................... 34
3.1.4.5.5 Attributes .................................................................. 34
3.1.4.5.6 Groups ...................................................................... 34
3.1.4.5.7 Attribute Groups ......................................................... 34
3.1.4.6 UninstallApp ................................................................. 34
3.1.4.6.1 Messages ................................................................... 35
3.1.4.6.1.1 UninstallAppSoapIn ............................................... 35
3.1.4.6.1.2 UninstallAppSoapOut .............................................. 35
3.1.4.6.2 Elements .................................................................. 36
1 Introduction

The Client Extension Web Service Protocol enables clients to use a web service to install, uninstall, and disable mail add-ins.

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:

**base64 encoding**: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable ASCII characters, as described in [RFC4648].

**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].

**mail add-in**: An Office Add-in that enhances an email or appointment item.

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

**Representational State Transfer (REST)**: A class of web services that is used to transfer domain-specific data by using HTTP, without additional messaging layers or session tracking, and returns textual data, such as XML.

**SOAP**: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. SOAP uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

**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.

**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 service**: A unit of application logic that provides data and services to other applications and can be called by using standard Internet transport protocols such as **HTTP**, Simple Mail Transfer Protocol (SMTP), or File Transfer Protocol (FTP). Web services can perform functions that range from simple requests to complicated business processes.

**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 operation**: A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

**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-OXWSCEXT] - v20210817
Client Extension Web Service Protocol
Copyright © 2021 Microsoft Corporation
Release: August 17, 2021
1.3 Overview

The Client Extension Web Service Protocol enables clients to use a web service to manage mail apps. It provides operations to install, uninstall, and disable mail apps. This protocol also provides operations to retrieve and manage the access token, the manifest file, and the landing page for a mail add-in.

1.4 Relationship to Other Protocols

This protocol is used by clients that implement support for Office Web Extensions, as described in [MS-OWEXML] and the Office Web Extensibility Manifest Format, as described in [MS-OWEMXML].

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 SOAP, as described in [SOAP1.1], to specify the structure information that is exchanged between the client and the server. This protocol uses the XML schema, as described in

[MS-OXWSCEXT] Microsoft Corporation, "Common Web Service Data Types".


This protocol uses SOAP over HTTP, as described in [RFC2616], and SOAP over HTTPS, as described in [RFC2818], as shown in the following layering diagram.

---

**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 operations that this protocol defines cannot be accessed unless the correct endpoint is identified in the HTTP web requests that target this protocol. This endpoint URL is returned by either the Autodiscover Publishing Lookup SOAP-Based Web Service Protocol, as specified in [MS-OXWSADISC], or the Autodiscover Publishing and Lookup Protocol, as specified in [MS-OXDSCLI].

### 1.6 Applicability Statement

This protocol is applicable to environments that use web services to implement client extensions for a user’s mailbox. This protocol is applicable to all SOAP-based clients, as described in [SOAP1.1].

### 1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports**: This protocol uses SOAP 1.1, as described in section 2.1.

- **Protocol Versions**: This protocol describes only one WSDL port type version. The RequestServerVersion element, as described in [MS-OXWSCDATA] section 2.2.3.9, identifies the WSDL port type version of the request. The ServerVersionInfo element, as described in [MS-OXWSCDATA] section 2.2.3.10, identifies the version of the server responding to the request.

- **Security and Authentication Methods**: This protocol relies on the web server that is hosting it to perform authentication.

- **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 Web Services Description Language (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

This protocol uses SOAP 1.1, as specified in [SOAP1.1].

This protocol MUST support SOAP over HTTP, as specified in [RFC2616]. The protocol SHOULD use secure communications by means of 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 specified in [XMLSCHEMA1/2] and [XMLSCHEMA2/2], and Web Services Description Language (WSDL), as specified in [WSDL].

2.2.1 Namespaces

This specification defines and references various XML namespaces 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/wsdl/soap/">http://schemas.xmlsoap.org/wsdl/soap/</a></td>
<td>[SOAP1.1]</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>
<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>wsd1</td>
<td><a href="http://schemas.xmlsoap.org/wsdl/">http://schemas.xmlsoap.org/wsdl/</a></td>
<td>[WSDL]</td>
</tr>
<tr>
<td>wsi</td>
<td><a href="http://ws-i.org/schemas/conformanceClaim/">http://ws-i.org/schemas/conformanceClaim/</a></td>
<td>[WSIBASIC]</td>
</tr>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>[XMLSCHEMA1/2] and [XMLSCHEMA2/2]</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
This specification does not define any common XML schema complex type definitions.

2.2.5 Simple Types
This specification does not define any common XML schema simple type definitions.

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

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

2.2.8 Attribute Groups
This specification does not define any common XML schema attribute group definitions.
3 Protocol Details

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

In the following sections, the schema definition might be less restrictive than the processing rules imposed by the protocol. The WSDL in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL specifies additional restrictions that reflect actual Microsoft product 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 and present.

3.1 ExchangeServicePortType Server Details

This protocol defines a single WSDL port type with operations that enable clients to install, uninstall, or disable mail apps.

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 operations as defined by this specification.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisableApp</td>
<td>The DisableApp WSDL operation disables a mail add-in.</td>
</tr>
<tr>
<td>GetAppManifests</td>
<td>The GetAppManifests WSDL operation retrieves the manifest files for a mail add-in.</td>
</tr>
<tr>
<td>GetAppMarketplaceUrl</td>
<td>The GetAppMarketplaceUrl WSDL operation retrieves the URL of the marketplace landing page for the mail add-in.</td>
</tr>
<tr>
<td>GetClientAccessToken</td>
<td>The GetClientAccessToken WSDL operation retrieves a client access token.</td>
</tr>
<tr>
<td>InstallApp</td>
<td>The InstallApp WSDL operation installs a mail add-in.</td>
</tr>
<tr>
<td>UninstallApp</td>
<td>The UninstallApp WSDL operation uninstalls a mail add-in.</td>
</tr>
</tbody>
</table>

3.1.4.1 DisableApp

The DisableApp WSDL operation disables a mail add-in.
The following is the WSDL port type specification of the DisableApp WSDL operation.

```
<wsdl:operation name="DisableApp" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    <wsdl:input message="tns:DisableAppSoapIn"/>
    <wsdl:output message="tns:DisableAppSoapOut"/>
</wsdl:operation>
```

The following is the WSDL binding specification of the DisableApp WSDL operation.

```
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <wsdl:input>
        <soap:header message="tns:DisableAppSoapIn" part="RequestVersion" use="literal"
            xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
            <soap:body use="literal" parts="request"
                xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
        </wsdl:input>

    <wsdl:output>
        <soap:body use="literal" parts="DisableAppResult"
            xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
            <soap:header message="tns:DisableAppSoapOut" part="ServerVersion" use="literal"
                xmlns:soap="http://schemas.xmlsoap.org/wsd..."
        </wsdl:output>
    </wsdl:operation>
```

The protocol client sends a DisableAppSoapIn request WSDL message and the protocol server responds with a DisableAppSoapOut 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</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisableAppSoapIn</td>
<td>Specifies a request to disable a mail add-in.</td>
</tr>
<tr>
<td>DisableAppSoapOut</td>
<td>Specifies the response to the DisableAppSoapIn request WSDL message.</td>
</tr>
</tbody>
</table>

### 3.1.4.1.1.1 DisableAppSoapIn

The DisableAppSoapIn request WSDL message specifies a request to disable a mail add-in.

The following is the DisableAppSoapIn WSDL message specification.

```
<wsdl:message name="DisableAppSoapIn" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="request" element="tns:DisableApp"/>
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
```


The parts of the DisableAppSoapIn WSDL message are described in the following table.
### 3.1.4.1.1.2 DisableAppSoapOut

The **DisableAppSoapOut** WSDL message specifies the response to the DisableApp request WSDL message.

The following is the **DisableAppSoapOut** WSDL message specification.

```xml
<wSDL:message name="DisableAppSoapOut" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/">
  <wSDL:part name="DisableAppResult" element="tns:DisableAppResponse"/>
  <wSDL:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wSDL:message>
```


The parts of the **DisableAppSoapOut** 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>DisableAppResult</td>
<td>DisableAppResponse (section 3.1.4.1.2.2)</td>
<td>Specifies the SOAP body of the response.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>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>

A successful **DisableApp** WSDL operation request returns a **DisableAppResponse** element with the **ResponseClass** ([MS-OXWSCDATA] section 2.2.4.67) attribute set to "Success". The **ResponseCode** ([MS-OXWSCDATA] section 2.2.4.67) element of the **DisableAppResponse** element is set to "NoError".

If the **DisableApp** WSDL operation is not successful, it returns a **DisableAppResponse** element with the **ResponseClass** attribute set to "Error". The **ResponseCode** element of the **DisableAppResponse** element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

### 3.1.4.1.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisableApp</td>
<td>The input data for the <strong>DisableApp</strong> WSDL operation.</td>
</tr>
</tbody>
</table>
### 3.1.4.1.2.1 DisableApp

The **DisableApp** element specifies the input data for the **DisableApp WSDL operation**.

```xml
<xs:element name="DisableApp" type="tns:DisableAppType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

### 3.1.4.1.2.2 DisableAppResponse

The **DisableAppResponse** element specifies the result data for the **DisableApp WSDL operation**.

```xml
<xs:element name="DisableAppResponse" type="tns:DisableAppResponseType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

### 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</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisableAppResponseType</td>
<td>Specifies the response to a DisableApp WSDL operation request.</td>
</tr>
<tr>
<td>DisableAppType</td>
<td>Specifies a request to disable a mail add-in.</td>
</tr>
</tbody>
</table>

### 3.1.4.1.3.1 DisableAppType

**Namespace:** http://schemas.microsoft.com/exchange/services/2006/messages

The **DisableAppType** complex type specifies a request to disable a **mail add-in**. This type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.17.

```xml
<xs:complexType name="DisableAppType"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="ID" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="DisableReason" type="t:DisableReasonType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

**ID:** An element of type **string**, as defined in [XMLSCHEMA2/2], that identifies the mail add-in.

**DisableReason:** An element of type **DisableReasonType** that specifies the reason to disable the mail add-in.
3.1.4.1.3.2 DisableAppResponseType

Namespace: http://schemas.microsoft.com/exchange/services/2006/messages

The DisableAppResponseType complex type specifies the response to a DisableApp WSDL operation request. This type extends the ResponseMessageType complex type, as specified in [MS-OXWSCDATA] section 2.2.4.67.

```xml
<xs:complexType name="DisableAppResponseType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
```

3.1.4.1.4 Simple Types

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

<table>
<thead>
<tr>
<th>Simple type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisableReasonType</td>
<td>Indicates the reason the mail add-in is being disabled.</td>
</tr>
</tbody>
</table>

3.1.4.1.4.1 DisableReasonType

Namespace: http://schemas.microsoft.com/exchange/services/2006/types

The DisableReasonType simple type indicates the reason the mail add-in is being disabled.

```xml
<xs:simpleType name="DisableReasonType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NoReason"/>
    <xs:enumeration value="OutlookClientPerformance"/>
    <xs:enumeration value="OWAClientPerformance"/>
    <xs:enumeration value="MobileClientPerformance"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the DisableReasonType simple type.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoReason</td>
<td>No reason specified.</td>
</tr>
<tr>
<td>OutlookClientPerformance</td>
<td>The mail add-in is being disabled from the messaging client due to performance.</td>
</tr>
<tr>
<td>OWAClientPerformance</td>
<td>The mail add-in is being disabled from the web client due to performance.</td>
</tr>
<tr>
<td>MobileClientPerformance</td>
<td>The mail add-in is being disabled from the mobile client due to performance.</td>
</tr>
</tbody>
</table>

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 GetAppManifests
The GetAppManifests WSDL operation retrieves the manifest files for a mail add-in.

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

```xml
<wsdl:operation name="GetAppManifests" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetAppManifestsSoapIn"/>
  <wsdl:output message="tns:GetAppManifestsSoapOut"/>
</wsdl:operation>
```

The following is the WSDL binding specification of the GetAppManifests WSDL operation.

```xml
<wsdl:operation name="GetAppManifests" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input>
      <soap:header message="tns:GetAppManifestsSoapIn" part="RequestVersion" use="literal" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
        <soap:body use="literal" parts="request" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
        </soap:body>
      </soap:header>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" parts="GetAppManifestsResult" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
        </soap:header>
      </wsdl:output>
    </wsdl:output>
  </soap:operation>
</wsdl:operation>
```

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</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAppManifestsSoapIn</td>
<td>Specifies a request to retrieve mail add-in manifest files.</td>
</tr>
<tr>
<td>GetAppManifestsSoapOut</td>
<td>Specifies the response to the GetAppManifestsSoapIn request WSDL message.</td>
</tr>
</tbody>
</table>

3.1.4.2.1.1 GetAppManifestsSoapIn
The GetAppManifestsSoapIn WSDL message specifies a request to retrieve mail add-in manifest files.

The following is the GetAppManifestsSoapIn WSDL message specification.
The **GetAppManifestsSoapIn** WSDL message is the input message for the SOAP action http://schemas.microsoft.com/exchange/services/2006/messages/GetAppManifests.

The parts of the **GetAppManifestsSoapIn** 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>request</td>
<td>GetAppManifests</td>
<td>Specifies the SOAP body of the request to retrieve manifest files.</td>
</tr>
<tr>
<td>RequestVersion</td>
<td>RequestServerVersion</td>
<td>Specifies a SOAP header that identifies the schema version for the <strong>GetAppManifests</strong> WSDL operation request.</td>
</tr>
</tbody>
</table>

### 3.1.4.2.1.2 GetAppManifestsSoapOut

The **GetAppManifestsSoapOut** WSDL message specifies the response to the **GetAppManifestsSoapIn** request WSDL message.

The following is the **GetAppManifestsSoapOut** WSDL message specification.

```
<wsl:message name="GetAppManifestsSoapOut" xmlns:wsl="http://schemas.xmlsoap.org/wsl/">
  <wsl:part name="GetAppManifestsResult" element="tns:GetAppManifestsResponse"/>
  <wsl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsl:message>
```

The **GetAppManifestsSoapOut** WSDL message is the output message for the SOAP action http://schemas.microsoft.com/exchange/services/2006/messages/GetAppManifests.

The parts of the **GetAppManifestsSoapOut** 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>GetAppManifestsResult</td>
<td>GetAppManifestsResponse</td>
<td>Specifies the SOAP body of the response.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>ServerVersionInfo</td>
<td>Specifies a SOAP header that identifies the server version for the response.</td>
</tr>
</tbody>
</table>

A successful **GetAppManifests WSDL operation** request returns a **GetAppManifestsResponse** element with the **ResponseClass** ([MS-OXWSCDATA] section 2.2.4.67) attribute set to "Success". The **ResponseCode** ([MS-OXWSCDATA] section 2.2.4.67) element of the **GetAppManifestsResponse** element is set to "NoError".

If the **GetAppManifests** WSDL operation is not successful, it returns a **GetAppManifestsResponse** element with the **ResponseClass** attribute set to "Error". The **ResponseCode** element of the
GetAppManifestsResponse element is set to one of the common errors defined in [MS-OXWSCEXT] section 2.2.5.24.

3.1.4.2.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAppManifests</td>
<td>The input data for the GetAppManifests WSDL operation.</td>
</tr>
<tr>
<td>GetAppManifestsResponse</td>
<td>The result data for the GetAppManifests WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.2.2.1 GetAppManifests

The GetAppManifests element specifies the input data for the GetAppManifests WSDL operation.

```xml
<xs:element name="GetAppManifests" type="tns:GetAppManifestsType" xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

3.1.4.2.2.2 GetAppManifestsResponse

The GetAppManifestsResponse element specifies the result data for the GetAppManifests WSDL operation.

```xml
<xs:element name="GetAppManifestsResponse" type="tns:GetAppManifestsResponseType" xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

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</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayOfAppManifestsType</td>
<td>Specifies an array of mail add-in manifest files.</td>
</tr>
<tr>
<td>GetAppManifestsResponseType</td>
<td>Specifies the response to a GetAppManifests WSDL operation request</td>
</tr>
<tr>
<td>GetAppManifestsType</td>
<td>Specifies a request to retrieve mail add-in manifest files.</td>
</tr>
</tbody>
</table>

3.1.4.2.3.1 GetAppManifestsType

Namespace: http://schemas.microsoft.com/exchange/services/2006/messages

The GetAppManifestsType complex type specifies a request to retrieve mail add-in manifest files. This type extends the BaseRequestType complex type, as specified in [MS-OXWSCDATA] section 2.2.4.17.

```xml
<xs:complexType name="GetAppManifestsType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestType"/>
  </xs:complexContent>
</xs:complexType>
```
3.1.4.2.3.2  GetAppManifestsResponseType

Namespace: http://schemas.microsoft.com/exchange/services/2006/messages

The GetAppManifestsResponseType complex type specifies the response to a GetAppManifests WSDL operation request. This type extends the ResponseMessageType, as specified in [MS-OXWSCDATA] section 2.2.4.67.

<xs:complexType name="GetAppManifestsResponseType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ResponseMessageType">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Manifests" type="tns:ArrayOfAppManifestsType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

Manifests: An element of type ArrayOfAppManifestsType that represents an array of mail add-in manifest files encoded with base64 encoding.

3.1.4.2.3.3  ArrayOfAppManifestsType

Namespace: http://schemas.microsoft.com/exchange/services/2006/messages

The ArrayOfAppManifestsType complex type specifies an array of mail add-in manifest files.

<xs:complexType name="ArrayOfAppManifestsType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Manifest" type="xs:base64Binary"/>
  </xs:sequence>
</xs:complexType>

Manifest: An element of type base64Binary, as defined in [XMLSCHEMA2/2], that represents a mail add-in manifest file encoded with base64 encoding.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.
3.1.4.3 GetAppMarketplaceUrl

The GetAppMarketplaceUrl WSDL operation retrieves the URL of the marketplace landing page for the mail add-in.

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

```xml
<wsdl:operation name="GetAppMarketplaceUrl" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input message="tns:GetAppMarketplaceUrlSoapIn"/>
    <wsdl:output message="tns:GetAppMarketplaceUrlSoapOut"/>
</wsdl:operation>
```

The following is the WSDL binding specification for the GetAppMarketplaceUrl WSDL operation.

```xml
<wsdl:operation name="GetAppMarketplaceUrl" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
        <wsdl:input>
            <soap:header message="tns:GetAppMarketplaceUrlSoapIn" part="RequestVersion" use="literal" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap"/>
            <soap:body use="literal" parts="request" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" parts="GetAppMarketplaceUrlResult" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap"/>
        </wsdl:output>
    </soap:operation>
</wsdl:operation>
```

The protocol client sends a GetAppMarketplaceUrlSoapIn request WSDL message and the protocol server responds with a GetAppMarketplaceUrlSoapOut response WSDL message.

3.1.4.3.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAppMarketplaceUrlSoapIn</td>
<td>Specifies a request to retrieve the marketplace URL.</td>
</tr>
<tr>
<td>GetAppMarketplaceUrlSoapOut</td>
<td>Specifies the response to the GetAppMarketPlaceUrlSoapIn request WSDL message.</td>
</tr>
</tbody>
</table>

3.1.4.3.1.1 GetAppMarketplaceUrlSoapIn

The GetAppMarketplaceUrlSoapIn WSDL message specifies a request to retrieve the marketplace URL.

The following is the GetAppMarketplaceUrlSoapIn WSDL message specification.

```xml
<wsdl:message name="GetAppMarketplaceUrlSoapIn"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="request" element="tns:GetAppMarketplaceUrl"/>
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
```

The parts of the `GetAppMarketplaceUrlSoapIn` 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>request</td>
<td><code>GetAppMarketplaceUrl</code> (section 3.1.4.3.2.1)</td>
<td>Specifies the SOAP body of the request to retrieve the marketplace URL.</td>
</tr>
<tr>
<td>RequestVersion</td>
<td><code>RequestServerVersion</code> ([MS-OXWSCDATA] section 2.2.3.9)</td>
<td>Specifies a SOAP header that identifies the schema version for the GetAppMarketplaceUrl WSDL operation request.</td>
</tr>
</tbody>
</table>

### 3.1.4.3.1.2 GetAppMarketplaceUrlSoapOut

The `GetAppMarketplaceUrlSoapOut` WSDL message specifies the response to a `GetAppMarketplaceUrlSoapIn` request WSDL message.

The following is the `GetAppMarketplaceUrlSoapOut` WSDL message specification.

```xml
<wsdl:message name="GetAppMarketplaceUrlSoapOut"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:part name="GetAppMarketplaceUrlResult" element="tns:GetAppMarketplaceUrlResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```


The parts of the `GetAppMarketplaceUrlSoapOut` 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>GetAppMarketplaceUrlResult</td>
<td><code>GetAppMarketplaceUrlResponse</code> (section 3.1.4.3.2.2)</td>
<td>Specifies the SOAP body of the response.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td><code>ServerVersionInfo</code> ([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>

A successful `GetAppMarketplaceUrl` WSDL operation request returns a `GetAppMarketplaceUrlResponse` element with the `ResponseClass` ([MS-OXWSCDATA] section 2.2.4.67) attribute set to "Success". The `ResponseCode` ([MS-OXWSCDATA] section 2.2.4.67) element of the `GetAppMarketplaceUrlResponse` element is set to "NoError".

If the `GetAppMarketplaceUrl` WSDL operation is not successful, it returns a `GetAppMarketplaceUrlResponse` element with the `ResponseClass` attribute set to "Error". The `ResponseCode` element of the `GetAppMarketplaceUrlResponse` element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.
3.1.4.3.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAppMarketplaceUrl</td>
<td>The input data for the GetAppMarketplaceUrl WSDL operation.</td>
</tr>
<tr>
<td>GetAppMarketplaceUrlResponse</td>
<td>The result data for the GetAppMarketplaceUrl WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.3.2.1 GetAppMarketplaceUrl

The GetAppMarketplaceUrl element specifies the input data for the GetAppMarketplaceUrl WSDL operation.

```xml
<x:s:element name="GetAppMarketplaceUrl" type="tns:GetAppMarketplaceUrlType"
xmlns:x="http://www.w3.org/2001/XMLSchema"/>
```

3.1.4.3.2.2 GetAppMarketplaceUrlResponse

The GetAppMarketplaceUrlResponse element specifies the result data for the GetAppMarketplaceUrl WSDL operation.

```xml
<x:s:element name="GetAppMarketplaceUrlResponse"
  type="tns:GetAppMarketplaceUrlResponseMessageType"
  xmlns:x="http://www.w3.org/2001/XMLSchema"/>
```

3.1.4.3.3 Complex Types

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAppMarketplaceUrlResponseMessageType</td>
<td>Specifies the response message for a GetAppMarketplaceUrl WSDL operation request.</td>
</tr>
<tr>
<td>GetAppMarketplaceUriType</td>
<td>Specifies a request to retrieve the marketplace URL.</td>
</tr>
</tbody>
</table>

3.1.4.3.3.1 GetAppMarketplaceUriType

**Namespace:** http://schemas.microsoft.com/exchange/services/2006/messages

The GetAppMarketplaceUriType complex type specifies a request to retrieve the marketplace URL for a mail add-in. This type extends the BaseRequestType complex type, as specified in [MS-OXWSCEXT] section 2.2.4.17.

```xml
<x:s:complexType name="GetAppMarketplaceUriType"
  xmlns:x="http://www.w3.org/2001/XMLSchema">
    <xs:complexContent mixed="false">
      <xs:extension base="tns:BaseRequestType"/>
    </xs:complexContent>
</x:s:complexType>
```
3.1.4.3.2  GetAppMarketplaceUrlResponseMessageType

Namespace: http://schemas.microsoft.com/exchange/services/2006/messages

The GetAppMarketplaceUrlResponseMessageType complex type specifies the response to a GetAppMarketplaceUrl WSDL operation request. This type extends the ResponseMessageType complex type, as specified in [MS-OXWSCDATA] section 2.2.4.67.

<AppMarketplaceUrl: An element of type string, as defined in [XMLSCHEMA2/2], that specifies the URL for the mail add-in.

3.1.4.3.4 Simple Types

None.

3.1.4.3.5 Attributes

None.

3.1.4.3.6 Groups

None.

3.1.4.3.7 Attribute Groups

None.

3.1.4.4 GetClientAccessToken

The GetClientAccessToken WSDL operation retrieves a client access token.

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

The following is the WSDL binding specification of the GetClientAccessToken WSDL operation.
The protocol client sends a `GetClientAccessTokenSoapIn` request **WSDL message** and the protocol server responds with a `GetClientAccessTokenSoapOut` response **WSDL message**.

### 3.1.4.4.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetClientAccessTokenSoapIn</code></td>
<td>Specifies a request to retrieve a client access token.</td>
</tr>
<tr>
<td><code>GetClientAccessTokenSoapOut</code></td>
<td>Specifies the response to a <code>GetClientAccessTokenSoapIn</code> request <strong>WSDL message</strong>.</td>
</tr>
</tbody>
</table>

#### 3.1.4.4.1.1 `GetClientAccessTokenSoapIn`

The `GetClientAccessTokenSoapIn` **WSDL message** specifies a request to retrieve a client access token.

The following is the `GetClientAccessTokenSoapIn` **WSDL message** specification.

```xml
<wsdl:message name="GetClientAccessTokenSoapIn"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <wsdl:part name="request" element="tns:GetClientAccessToken"/>
    <wsdl:part name="RequestVersion" element="tns:RequestServerVersion"/>
</wsdl:message>
```

The `GetClientAccessTokenSoapIn` **WSDL message** is the input message for the **SOAP action** `http://schemas.microsoft.com/exchange/services/2006/messages/GetClientAccessToken`.

The parts of the `GetClientAccessTokenSoapIn` **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>request</td>
<td><code>GetClientAccessToken</code> (section 3.1.4.4.2.1)</td>
<td>Specifies the <strong>SOAP body</strong> of the request to retrieve the client access token.</td>
</tr>
<tr>
<td>RequestVersion</td>
<td><code>RequestServerVersion</code> ([MS-OXWSCDATA] section)</td>
<td>Specifies a <strong>SOAP header</strong> that identifies the schema version for the <code>GetClientAccessToken</code> <strong>WSDL operation</strong> request.</td>
</tr>
</tbody>
</table>
### 3.1.4.4.1.2 GetClientAccessTokenSoapOut

The GetClientAccessTokenSoapOut WSDL message specifies the response to a GetClientAccessToken WSDL operation request.

The following is the GetClientAccessTokenSoapOut WSDL message specification.

```xml
<wSDL:message name="GetClientAccessTokenSoapOut" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:part name="GetClientAccessTokenResult" element="tns:GetClientAccessTokenResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```


The parts of the GetClientAccessTokenSoapOut 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>GetClientAccessTokenResult</td>
<td>GetClientAccessTokenResponse</td>
<td>Specifies the SOAP body of the response that contains the requested information.</td>
</tr>
<tr>
<td></td>
<td>(section 3.1.4.4.2.2)</td>
<td></td>
</tr>
<tr>
<td>ServerVersion</td>
<td>ServerVersionInfo</td>
<td>Specifies a SOAP header that identifies the server version for the response.</td>
</tr>
<tr>
<td></td>
<td>([MS-OXWSCEXT] section 2.2.3.10)</td>
<td></td>
</tr>
</tbody>
</table>

A successful GetClientAccessToken WSDL operation request returns a GetClientAccessTokenResponse element with the ResponseClass ([MS-OXWSCEXT] section 2.2.4.67) attribute set to "Success". The ResponseCode ([MS-OXWSCEXT] section 2.2.4.67) element of the GetClientAccessTokenResponse element is set to "NoError".

If the GetClientAccessToken WSDL operation is not successful, it returns a GetClientAccessTokenResponse element with the ResponseClass attribute set to "Error". The ResponseCode element of the GetClientAccessTokenResponse element is set to one of the common errors defined in [MS-OXWSCEXT] section 2.2.5.24.

### 3.1.4.4.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetClientAccessToken</td>
<td>The input data for the GetClientAccessToken WSDL operation.</td>
</tr>
<tr>
<td>GetClientAccessTokenResponse</td>
<td>The result data for the GetClientAccessToken WSDL operation.</td>
</tr>
</tbody>
</table>
3.1.4.4.2.1 GetClientAccessToken

The GetClientAccessToken element specifies the input data for the GetClientAccessToken WSDL operation.

```
<xs:element name="GetClientAccessToken" type="tns:GetClientAccessTokenType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

3.1.4.4.2.2 GetClientAccessTokenResponse

The GetClientAccessTokenResponse element specifies the result data for the GetClientAccessToken WSDL operation.

```
<xs:element name="GetClientAccessTokenResponse" type="tns:GetClientAccessTokenResponseType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

3.1.4.4.3 Complex Types

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientAccessTokenRequestType</td>
<td>Identifies the request type for a client access token.</td>
</tr>
<tr>
<td>ClientAccessTokenType</td>
<td>Specifies a client access token.</td>
</tr>
<tr>
<td>GetClientAccessTokenResponseMessageType</td>
<td>Specifies a response message for a GetClientAccessToken WSDL operation request.</td>
</tr>
<tr>
<td>GetClientAccessTokenResponseType</td>
<td>Specifies the response to a GetClientAccessToken WSDL operation request.</td>
</tr>
<tr>
<td>GetClientAccessTokenType</td>
<td>Specifies a request to retrieve a client access token.</td>
</tr>
<tr>
<td>NonEmptyArrayOfClientAccessTokenRequestsType</td>
<td>Specifies an array of client access token requests.</td>
</tr>
</tbody>
</table>

3.1.4.4.3.1 GetClientAccessTokenType

**Namespace:** http://schemas.microsoft.com/exchange/services/2006/messages

The GetClientAccessTokenType complex type specifies a request to retrieve a client access token. This type extends the BaseRequestType complex type, as specified in [MS-OXWSCEXT] section 2.2.4.17.

```
<xs:complexType name="GetClientAccessTokenType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="TokenRequests" type="t:NonEmptyArrayOfClientAccessTokenRequestsType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```
TokenRequests: An element of type NonEmptyArrayOfClientAccessTokenRequestsType that specifies an array of client access token requests.

3.1.4.4.3.2 NonEmptyArrayOfClientAccessTokenRequestsType

Namespace: http://schemas.microsoft.com/exchange/services/2006/types

The NonEmptyArrayOfClientAccessTokenRequestsType complex type specifies an array of client access token requests.

<xs:complexType name="NonEmptyArrayOfClientAccessTokenRequestsType"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" name="TokenRequest"
          type="t:ClientAccessTokenRequestType"/>
  </xs:sequence>
</xs:complexType>

TokenRequest: An element of type ClientAccessTokenRequestType that represents a single token request.

3.1.4.4.3.3 ClientAccessTokenRequestType

Namespace: http://schemas.microsoft.com/exchange/services/2006/types

The ClientAccessTokenRequestType complex type identifies the request type for a client access token.

<xs:complexType name="ClientAccessTokenRequestType"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element name="Id" type="xs:string"/>
    <xs:element name="TokenType" type="t:ClientAccessTokenTypeType"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Scope" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

Id: An element of type string, as defined in [XMLSCHEMA2/2], that contains the identifier of the mail add-in.

TokenType: An element of type ClientAccessTokenTypeType that identifies the type of token that will be returned in the response.

Scope: An element of type string that specifies the access allowed by the token. <1>

3.1.4.4.3.4 GetClientAccessTokenResponseMessageType

Namespace: http://schemas.microsoft.com/exchange/services/2006/messages

The GetClientAccessTokenResponseMessageType complex type specifies a response message for a GetClientAccessToken WSDL operation request. This type extends the ResponseMessageType complex type, as specified in [MS-OXWSCDATA] section 2.2.4.67.
<xs:complexType name="GetClientAccessTokenResponseMessageType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ResponseMessageType">
      <xs:sequence>
        <xs:element minOccurs="0" name="Token" type="t:ClientAccessTokenType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

**Token:** An element of type **ClientAccessTokenType** that represents the client access token.

### 3.1.4.4.3.5 ClientAccessTokenType

**Namespace:** http://schemas.microsoft.com/exchange/services/2006/types

The **ClientAccessTokenType** complex type specifies a client access token.

<xs:complexType name="ClientAccessTokenType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element name="Id" type="xs:string"/>
    <xs:element name="TokenType" type="t:ClientAccessTokenTypeType"/>
    <xs:element name="TokenValue" type="xs:string"/>
    <xs:element name="TTL" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>

**Id:** An element of type **string**, as defined in [XMLSCHEMA2/2], that represents a client access token identifier.

**TokenType:** An element of type **ClientAccessTokenTypeType** that represents the type of the token.

**TokenValue:** An element of type **string** that represents the value of the token.

**TTL:** An element of type **integer**, as defined in [XMLSCHEMA2/2], that represents the time to live for the token.

### 3.1.4.4.3.6 GetClientAccessTokenResponseType

**Namespace:** http://schemas.microsoft.com/exchange/services/2006/messages

The **GetClientAccessTokenResponse** complex type specifies the response to a **GetClientAccessToken** WSDL operation request. This type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.18.

<xs:complexType name="GetClientAccessTokenResponseMessageType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>

### 3.1.4.4.4 Simple Types

The following table summarizes the **XML schema** simple type definitions that are specific to this operation.
### 3.1.4.4.4.1 ClientAccessTokenTypeType

**Namespace:** http://schemas.microsoft.com/exchange/services/2006/types

The **ClientAccessTokenTypeType** simple type specifies the type of a client access token.

```xml
<xs:simpleType name="ClientAccessTokenTypeType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="CallerIdentity"/>
    <xs:enumeration value="ExtensionCallback"/>
    <xs:enumeration value="ScopedToken"/>
    <xs:enumeration value="ExtensionRestApiCallback"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **ClientAccessTokenTypeType** simple type.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallerIdentity</td>
<td>Indicates a caller identity client access token.</td>
</tr>
<tr>
<td>ExtensionCallback</td>
<td>Indicates an extension callback client access token.</td>
</tr>
<tr>
<td>ScopedToken</td>
<td>Indicates a token with limited scope.</td>
</tr>
<tr>
<td>ExtensionRestApiCallback</td>
<td>Indicates an extension REST API callback client access token.</td>
</tr>
</tbody>
</table>

### 3.1.4.4.5 Attributes

None.

### 3.1.4.4.6 Groups

None.

### 3.1.4.4.7 Attribute Groups

None.

### 3.1.4.5 InstallApp

The **InstallApp** WSDL operation installs a mail add-in.

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

```xml
<wsdl:operation name="InstallApp" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:InstallAppStateIn"/>
  <wsdl:output message="tns:InstallAppStateOut"/>
</wsdl:operation>
```
The following is the WSDL binding specification of the **InstallApp** WSDL operation.

```xml
<wSDL:operation name="InstallApp" xmlns:wSDL="http://schemas.xmlsoap.org/wsd1/">
  <soap:operation
    soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/InstallApp"
    xmlns:soap="http://schemas.xmlsoap.org/wsd1/">
    <wsdl:input>
      <soap:header
        message="tns:InstallAppSoapIn" part="RequestVersion" use="literal"
        xmlns:soap="http://schemas.xmlsoap.org/wsd1/">
      </soap:header>
      <soap:body use="literal" parts="request"
        xmlns:soap="http://schemas.xmlsoap.org/wsd1/">
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" parts="InstallAppResult"
        xmlns:soap="http://schemas.xmlsoap.org/wsd1/">
      </soap:body>
      <soap:header
        message="tns:InstallAppSoapOut" part="ServerVersion" use="literal"
        xmlns:soap="http://schemas.xmlsoap.org/wsd1/">
      </soap:header>
    </wsdl:output>
  </soap:operation>
</wSDL:operation>
```

### 3.1.4.5.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstallAppSoapIn</td>
<td>Specifies a request to install a mail add-in.</td>
</tr>
<tr>
<td>InstallAppSoapOut</td>
<td>Specifies the response to the InstallAppSoapIn request WSDL message.</td>
</tr>
</tbody>
</table>

### 3.1.4.5.1.1 InstallAppSoapIn

The **InstallAppSoapIn** **WSDL message** specifies a request to install a mail add-in.

The following is the **InstallAppSoapIn** WSDL message specification.

```xml
<wSDL:message name="InstallAppSoapIn" xmlns:wSDL="http://schemas.xmlsoap.org/wsd1/">
  <wSDL:part name="request" element="tns:InstallApp"/>
  <wSDL:part name="RequestVersion" element="t:RequestServerVersion"/>
</wSDL:message>
```

The **InstallAppSoapIn** WSDL message is the input message for the **SOAP action** http://schemas.microsoft.com/exchange/services/2006/messages/InstallApp.

The parts of the **InstallAppSoapIn** 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>request</td>
<td><strong>InstallApp</strong> (section 3.1.4.5.2.1)</td>
<td>Specifies the <strong>SOAP body</strong> of the request to install a mail add-in.</td>
</tr>
<tr>
<td>RequestVersion</td>
<td>RequestServerVersion ([MS-OXWSCEXT] section 2.2.3.9)</td>
<td>Specifies a <strong>SOAP header</strong> that identifies the schema version for the InstallApp WSDL operation request.</td>
</tr>
</tbody>
</table>
3.1.4.5.1.2 InstallAppSoapOut

The InstallAppSoapOut WSDL message specifies the response to the InstallAppSoapIn request.

The following is the InstallAppSoapOut WSDL message specification.

```
<wsdl:message name="InstallAppSoapOut" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:part name="InstallAppResult" element="tns:InstallAppResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

The InstallAppSoapOut WSDL message is the output message for the SOAP action http://schemas.microsoft.com/exchange/services/2006/messages/InstallApp.

The parts of the InstallAppSoapOut 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>InstallAppResult</td>
<td>InstallAppResponse (section 3.1.4.5.2.2)</td>
<td>Specifies the SOAP body of the response.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>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>

A successful InstallApp WSDL operation request returns an InstallAppResponse element with the ResponseClass ([MS-OXWSCDATA] section 2.2.4.67) attribute set to "Success". The ResponseCode ([MS-OXWSCDATA] section 2.2.4.67) element of the InstallAppResponse element is set to "NoError".

If the InstallApp WSDL operation is not successful, it returns an InstallAppResponse element with the ResponseClass attribute set to "Error". The ResponseCode element of the InstallAppResponse element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.5.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstallApp</td>
<td>The input data for the InstallApp WSDL operation.</td>
</tr>
<tr>
<td>InstallAppResponse</td>
<td>The result data for the InstallApp WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.5.2.1 InstallApp

The InstallApp element specifies the input data for the InstallApp WSDL operation.

```
<xs:element name="InstallApp" type="tns:InstallAppType"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```
3.1.4.5.2.2 InstallAppResponse

The **InstallAppResponse** element specifies the result data for the **InstallApp WSDL operation**.

```xml
<xs:element name="InstallAppResponse" type="tns:InstallAppResponseType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

3.1.4.5.3 Complex Types

The following table summarizes the [XML schema](https://www.w3.org/2001/XMLSchema) complex type definitions that are specific to this operation.

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstallAppResponseType</td>
<td>Specifies the response to an <strong>InstallApp WSDL operation</strong> request.</td>
</tr>
<tr>
<td>InstallAppType</td>
<td>Specifies a request to install a mail add-in.</td>
</tr>
</tbody>
</table>

3.1.4.5.3.1 InstallAppType

**Namespace:** http://schemas.microsoft.com/exchange/services/2006/messages

The **InstallAppType** complex type specifies a request to install a mail add-in. This type extends the **BaseRequestType** complex type, as specified in **[MS-OXWSCDATA]** section 2.2.4.17.

```xml
<xs:complexType name="InstallAppType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="Manifest" type="xs:base64Binary"/>
        <xs:element minOccurs="0" maxOccurs="1" name="MarketplaceAssetId" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="MarketplaceContentMarket" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="SendWelcomeEmail" type="xs:boolean"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

**Manifest:** An element of type **base64Binary**, as defined in **[XMLSCHEMA2/2]**, that contains the mail add-in manifest file encoded with **base64 encoding**.

**MarketplaceAssetId:** An element of type **string**, as defined in **[XMLSCHEMA2/2]**, that specifies the asset id of the add-in. <3>

**MarketplaceContentMarket:** An element of type **string** that specifies the target market for content. <4>

**SendWelcomeEmail:** An element of type **boolean**, as defined in **[XMLSCHEMA2/2]**, that specifies whether to send a welcome email for the add-in. <5>

3.1.4.5.3.2 InstallAppResponseType

**Namespace:** http://schemas.microsoft.com/exchange/services/2006/messages
The InstallAppResponseType complex type specifies the response to an InstallApp WSDL operation request. This type extends the ResponseMessageType complex type, as specified in [MS-OXWSCDATA] section 2.2.4.67.

```xml
<xs:complexType name="InstallAppResponseType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ResponseMessageType">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="WasFirstInstall" type="xs:boolean"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

**WasFirstInstall:** An element of type `boolean`, as defined in [XMLSCHEMA2/2], that specifies whether this is the first time the app is being installed. <6>

### 3.1.4.5.4 Simple Types

None.

### 3.1.4.5.5 Attributes

None.

### 3.1.4.5.6 Groups

None.

### 3.1.4.5.7 Attribute Groups

None.

### 3.1.4.6 UninstallApp

The UninstallApp WSDL operation uninstalls a mail add-in.

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

```xml
<wsdl:operation name="UninstallApp" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:UninstallAppSoapIn"/>
  <wsdl:output message="tns:UninstallAppSoapOut"/>
</wsdl:operation>
```

The following is the WSDL binding specification of the UninstallApp WSDL operation.

```xml
<soap:operation name="UninstallApp" xmlns:soap="http://schemas.xmlsoap.org/soap/">
    <soap:header message="tns:UninstallAppSoapIn" part="RequestVersion" use="literal" xmlns:soap="http://schemas.xmlsoap.org/soap/">
      <soap:body use="literal" parts="request" xmlns:soap="http://schemas.xmlsoap.org/soap/">
    </soap:input>
  </soap:input>
  <soap:output>
    <soap:body use="literal" parts="UninstallAppResult" xmlns:soap="http://schemas.xmlsoap.org/soap/">
  </soap:output>
</soap:operation>
```
The protocol client sends an UninstallAppSoapIn request WSDL message and the protocol server responds with an UninstallAppSoapOut response WSDL message.

### 3.1.4.6.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UninstallAppSoapIn</td>
<td>Specifies a request to uninstall a mail add-in.</td>
</tr>
<tr>
<td>UninstallAppSoapOut</td>
<td>Specifies the response to the UninstallAppSoapIn request WSDL message.</td>
</tr>
</tbody>
</table>

#### 3.1.4.6.1.1 UninstallAppSoapIn

The UninstallAppSoapIn WSDL message specifies a request to uninstall a mail add-in.

The following is the UninstallAppSoapIn WSDL message specification.

```xml
<wsdl:message name="UninstallAppSoapIn" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="request" element="tns:UninstallApp"/>
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
```

The UninstallAppSoapIn WSDL message is the input message for the SOAP action http://schemas.microsoft.com/exchange/services/2006/messages/UninstallApp.

The parts of the UninstallAppSoapIn 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>request</td>
<td>UninstallApp</td>
<td>Specifies the SOAP body of the request to uninstall a mail add-in.</td>
</tr>
<tr>
<td></td>
<td>(section 3.1.4.6.2.1)</td>
<td></td>
</tr>
<tr>
<td>RequestVersion</td>
<td>RequestServerVersion</td>
<td>Specifies a SOAP header that identifies the schema version for the UninstallApp WSDL operation request.</td>
</tr>
<tr>
<td></td>
<td>([MS-OXWSCDATA] section 2.2.3.9)</td>
<td></td>
</tr>
</tbody>
</table>

#### 3.1.4.6.1.2 UninstallAppSoapOut

The UninstallAppSoapOut WSDL message specifies the response to the UninstallAppSoapIn request WSDL message.

The following is the UninstallAppSoapOut WSDL message specification.

```xml
<wsdl:message name="UninstallAppSoapOut" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="UninstallAppResult" element="tns:UninstallAppResponse"/>
</wsdl:message>
```
The **UninstallAppSoapOut** WSDL message is the output message for the **SOAP action** http://schemas.microsoft.com/exchange/services/2006/messages/UninstallApp.

The parts of the **UninstallAppSoapOut** 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>UninstallAppResult</td>
<td>UninstallAppResponse (section 3.1.4.6.2.2)</td>
<td>Represents the result of the operation.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.10)</td>
<td>Specifies a <strong>SOAP header</strong> that identifies the server version for the response.</td>
</tr>
</tbody>
</table>

A successful **UninstallApp** **WSDL operation** request returns an **UninstallAppResponse** element with the **ResponseClass** ([MS-OXWSCDATA] section 2.2.4.67) attribute set to "Success". The **ResponseCode** ([MS-OXWSCDATA] section 2.2.4.67) element of the **UninstallAppResponse** element is set to "NoError".

If the **UninstallApp** WSDL operation is not successful, it returns an **UninstallAppResponse** element with the **ResponseClass** attribute set to "Error". The **ResponseCode** element of the **UninstallAppResponse** element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

### 3.1.4.6.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UninstallApp</td>
<td>The input data for the <strong>UninstallApp</strong> <strong>WSDL operation</strong>.</td>
</tr>
<tr>
<td>UninstallAppResponse</td>
<td>The result data for the <strong>UninstallApp</strong> <strong>WSDL operation</strong>.</td>
</tr>
</tbody>
</table>

### 3.1.4.6.2.1 UninstallApp

The **UninstallApp** element specifies the input data for the **UninstallApp** **WSDL operation**.

```xml
<xs:element name="UninstallApp" type="tns:UninstallAppType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

### 3.1.4.6.2.2 UninstallAppResponse

The **UninstallAppResponse** element specifies the result data for the **UninstallApp** **WSDL operation**.
3.1.4.6.3 Complex Types

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UninstallAppResponseType</td>
<td>Specifies the response to an UninstallApp WSDL operation request.</td>
</tr>
<tr>
<td>UninstallAppType</td>
<td>Specifies a request to uninstall a mail add-in.</td>
</tr>
</tbody>
</table>

3.1.4.6.3.1 UninstallAppType

Namespace: http://schemas.microsoft.com/exchange/services/2006/messages

The UninstallAppType complex type specifies a request to uninstall a mail add-in. This type extends the BaseRequestType, as specified in [MS-OXWSCDATA] section 2.2.4.17.

```xml
<xs:complexType name="UninstallAppType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="ID" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

**ID:** An element of type string, as defined in [XMLSCHEMA2/2], that specifies the identifier of the mail add-in.

3.1.4.6.3.2 UninstallAppResponseType

Namespace: http://schemas.microsoft.com/exchange/services/2006/messages

The UninstallAppResponseType complex type specifies the response to an UninstallApp WSDL operation request. This type extends the ResponseMessageType complex type, as specified in [MS-OXWSCDATA] section 2.2.4.67.

```xml
<xs:complexType name="UninstallAppResponseType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
```

3.1.4.6.4 Simple Types

None.

3.1.4.6.5 Attributes

None.
3.1.4.6.6 Groups

None.

3.1.4.6.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.
4 Protocol Examples

4.1 Getting App Manifest Files

This example of the **GetAppManifests** operation, as defined in section 3.1.4.2, shows how a client can retrieve the manifest files for mail add-ins from the server. The client sends the following request.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
    <soap:Header>
        <t:RequestServerVersion Version="Exchange2013"/>
        <t:MailboxCulture>en-US</t:MailboxCulture>
        <t:TimeZoneContext>
            <t:TimeZoneDefinition Id="GMT Standard Time"/>
        </t:TimeZoneContext>
    </soap:Header>
    <soap:Body>
        <m:GetAppManifests/>
    </soap:Body>
</soap:Envelope>
```

The server sends the following successful response to the client. The response returns four manifest files. Each file is an XML file encoded with **base64 encoding**. The contents of the files have been abbreviated for readability.

```xml
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <s:Header>
        <s:ServerVersionInfo MajorVersion="15"
            MinorVersion="0"
            MajorBuildNumber="556"
            MinorBuildNumber="14"
            Version="Exchange2013"
            xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
            xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
    </s:Header>
    <s:Body xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <GetAppManifestsResponse ResponseClass="Success"
            xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
            <ResponseCode>NoError</ResponseCode>
            <m:Manifests xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
                <Manifest>PE9mZmljZUFwczBmN0x0aH</Manifest>
                <Manifest>PE9mZmljZUFwczBmN0x0aH</Manifest>
                <Manifest>PE9mZmljZUFwczBmN0x0aH</Manifest>
                <Manifest>PE9mZmljZUFwczBmN0x0aH</Manifest>
            </m:Manifests>
        </GetAppManifestsResponse>
    </s:Body>
</s:Envelope>
```
4.2 Getting a Client Access Token

This example of the GetClientAccessToken operation, as defined in section 3.1.4.4, shows how the client retrieves a client access token for a mail add-in specified by the Id element of the TokenRequest element, as described in section 3.1.4.4.3.3. The ID for the app is in the manifest file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
    <soap:Header>
        <t:RequestServerVersion Version="Exchange2013" />
    </soap:Header>
    <soap:Body>
        <m:GetClientAccessToken>
            <m:TokenRequests>
                <t:TokenRequest>
                    <t:Id>1C50226D-04B5-4AB2-9FCD-42E236B59E4B</t:Id>
                    <t:TokenType>CallerIdentity</t:TokenType>
                </t:TokenRequest>
            </m:TokenRequests>
        </m:GetClientAccessToken>
    </soap:Body>
</soap:Envelope>
```

The server sends the following successful response to the client. The token value has been abbreviated for readability.

```xml
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Header>
        <h:ServerVersionInfo MajorVersion="15"
            MinorVersion="0"
            MajorBuildNumber="556"
            MinorBuildNumber="14"
            Version="Exchange2013"
            xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
            xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
    </s:Header>
    <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <m:GetClientAccessTokenResponse xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
            xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
            <m:ResponseMessages>
                <m:GetClientAccessTokenResponseMessage ResponseClass="Success">
                    <m:ResponseCode>NoError</m:ResponseCode>
                    <m:Token>
                        <t:Id>1C50226D-04B5-4AB2-9FCD-42E236B59E4B</t:Id>
                        <t:TokenType>CallerIdentity</t:TokenType>
                        <t:TokenValue>eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6ImJRQnI5a3gwQ3plblFuaTFDdzJ6NWt1amJvYyJ9.</t:TokenValue>
                        <t:TTL>479</t:TTL>
                    </m:Token>
                </m:GetClientAccessTokenResponseMessage>
            </m:ResponseMessages>
        </m:GetClientAccessTokenResponse>
    </s:Body>
</s:Envelope>
```
4.3 Getting an App Marketplace URL

This example of the `GetAppMarketplaceUrl` operation, as defined in section 3.1.4.3, shows how the client can retrieve the URL of the marketplace landing page for the mail add-in.

```xml
<xml version="1.0" encoding="UTF-8"/>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
   xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2013"/>
    <t:MailboxCulture>en-US</t:MailboxCulture>
    <t:TimeZoneContext>
      <t:TimeZoneDefinition Id="GMT Standard Time"/>
    </t:TimeZoneContext>
  </soap:Header>
  <soap:Body>
    <m:GetAppMarketplaceUrl/>
  </soap:Body>
</soap:Envelope>
```

The server sends the following successful response to the client.

```xml
<xml version="1.0" encoding="utf-8"/>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <t:ServerVersionInfo MajorVersion="15"
      MinorVersion="0"
      MajorBuildNumber="556"
      MinorBuildNumber="14"
      Version="Exchange2013"
      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">
      <GetAppMarketplaceUrlResponse ResponseClass="Success"
       xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
        <ResponseCode>NoError</ResponseCode>
        <AppMarketplaceUrl>
          http://office15.officeredir.microsoft.com/rdidMktplcExchRedirect?app=outlook.exe&amp;ver=15&amp;clid=1033&amp;p1=15d0d56d14&amp;p2=44&amp;p3=0&amp;p4=HP&amp;p5=0&amp;Scop e=1&amp;CallbackURL=https%3a%2f%2fch1pr03.outlook.com%2fequip%2fExtension%2finstallFromURL.slab %3fexsvurl%3d1%26realm%3dmicrosoft.com</AppMarketplaceUrl>
    </GetAppMarketplaceUrlResponse>
  </soap:Header>
</soap:Envelope>
```

4.4 Installing a Mail App

This example of the `InstallApp` operation, as defined in section 3.1.4.5, shows how the client sends a request to the server to install a mail add-in. The manifest file of the mail add-in has been abbreviated for readability.

```xml
<xml version="1.0" encoding="UTF-8"/>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2013"/>
  </soap:Header>
</soap:Envelope>
```
The server sends the following successful response to the client.

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
   xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2013" />
    <t:MailboxCulture>en-US</t:MailboxCulture>
    <t:TimeZoneContext>
      <t:TimeZoneDefinition Id="GMT Standard Time"/>
    </t:TimeZoneContext>
  </soap:Header>
  <soap:Body>
    <m:InstallApp>
      <m:Manifest>PE9mZmljZUFwcCB4bWxucz0iaHR0cDovL3NjaG=</m:Manifest>
    </m:InstallApp>
  </soap:Body>
</soap:Envelope>

4.5 Disabling a Mail App

This example of the DisableApp operation, as defined in section 3.1.4.1, shows how the client sends a request to the server to disable a mail add-in. The ID of the app is in the manifest file.

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
   xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2013" />
    <t:MailboxCulture>en-US</t:MailboxCulture>
    <t:TimeZoneContext>
      <t:TimeZoneDefinition Id="GMT Standard Time"/>
    </t:TimeZoneContext>
  </soap:Header>
  <soap:Body>
    <m:DisableApp>
      <m:ID>1C50226D-04B5-4AB2-9FCD-42E236B59E4B</m:ID>
      <m:DisableReason>NoReason</m:DisableReason>
    </m:DisableApp>
  </soap:Body>
</soap:Envelope>
The server sends the following successful response 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="15"
    MinorVersion="0"
    MajorBuildNumber="556"
    MinorBuildNumber="14"
    Version="Exchange2013"
    xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
 </s:Header>
 <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <DisableAppResponse ResponseClass="Success"
    xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
     <ResponseCode>NoError</ResponseCode>
   </DisableAppResponse>
 </s:Body>
</s:Envelope>
```

4.6 Uninstalling a Mail App

This example of the `UninstallApp` operation, as defined in section 3.1.4.6, shows how the client sends a request to the server to uninstall a mail add-in. The ID of the app is in the manifest file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
 <soap:Header>
   <t:RequestServerVersion Version="Exchange2013" />
 </soap:Header>
 <soap:Body>
   <m:UninstallApp>
    <m:ID>1C50226D-04B5-4AB2-9FCD-42E236B59E4B</m:ID>
   </m:UninstallApp>
 </soap:Body>
</soap:Envelope>
```

The server sends the following successful response 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="15"
    MinorVersion="0"
    MajorBuildNumber="556"
    MinorBuildNumber="14"
    Version="Exchange2013"
    xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
 </s:Header>
 <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <UninstallAppResponse ResponseClass="Success"
    xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
    <ResponseCode>NoError</ResponseCode>
   </UninstallAppResponse>
 </s:Body>
</s:Envelope>
```
</UninstallAppResponse>
</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

For ease of implementation, the full WSDL is provided in this appendix.

```xml
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
    targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:types>
        <xs:schema elementFormDefault="qualified"
            targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
            version="Exchange2016" id="messages" xmlns:xs="http://www.w3.org/2001/XMLSchema">
            <xs:include schemaLocation="MS-OWWSCEXT-messages.xsd"/>
        </xs:schema>
    </wsdl:types>
    <wsdl:portType name="ExchangeServicePortType">
        <wsdl:operation name="DisableApp">
            <wsdl:input message="tns:DisableAppSoapIn"/>
            <wsdl:output message="tns:DisableAppSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="GetAppManifests">
            <wsdl:input message="tns:GetAppManifestsSoapIn"/>
            <wsdl:output message="tns:GetAppManifestsSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="GetAppMarketplaceUrl">
            <wsdl:input message="tns:GetAppMarketplaceUrlSoapIn"/>
            <wsdl:output message="tns:GetAppMarketplaceUrlSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="GetClientAccessToken">
            <wsdl:input message="tns:GetClientAccessTokenSoapIn"/>
            <wsdl:output message="tns:GetClientAccessTokenSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="InstallApp">
            <wsdl:input message="tns:InstallAppSoapIn"/>
            <wsdl:output message="tns:InstallAppSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="UninstallApp">
            <wsdl:input message="tns:UninstallAppSoapIn"/>
            <wsdl:output message="tns:UninstallAppSoapOut"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
        <wsdl:documentation>
            <ws:i:Claim conformsTo="http://ws-i.org/profiles/basic/1.0" xmlns:ws:i="http://ws-i.org/schemas/conformanceClaim"/>
        </wsdl:documentation>
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="DisableApp">
            <ws:input/>
            <soap:header message="tns:DisableAppSoapIn" part="RequestVersion" use="literal"/>
            <soap:body use="literal" parts="request"/>
            <ws:input/>
            <soap:output/>
            <soap:header message="tns:DisableAppSoapOut" part="ServerVersion" use="literal"/>
            <ws:output/>
        </wsdl:operation>
        <wsdl:operation name="GetClientAccessToken">
            <ws:input/>
        </wsdl:operation>
    </wsdl:binding>
</wsdl:definitions>
```
<soap:header message="tns:GetClientAccessTokenSoapIn" part="RequestVersion" use="literal"/>
<soap:body use="literal" parts="request"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal" parts="GetClientAccessTokenResult"/>
<soap:header message="tns:GetClientAccessTokenSoapOut" part="ServerVersion" use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetAppManifests">
<wsdl:input>
<soap:header message="tns:GetAppManifestsSoapIn" part="RequestVersion" use="literal"/>
<soap:body use="literal" parts="request"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal" parts="GetAppManifestsResult"/>
<soap:header message="tns:GetAppManifestsSoapOut" part="ServerVersion" use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetAppMarketplaceUrl">
<wsdl:input>
<soap:header message="tns:GetAppMarketplaceUrlSoapIn" part="RequestVersion" use="literal"/>
<soap:body use="literal" parts="request"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal" parts="GetAppMarketplaceUrlResult"/>
<soap:header message="tns:GetAppMarketplaceUrlSoapOut" part="ServerVersion" use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="InstallApp">
<wsdl:input>
<soap:header message="tns:InstallAppSoapIn" part="RequestVersion" use="literal"/>
<soap:body use="literal" parts="request"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal" parts="InstallAppResult"/>
<soap:header message="tns:InstallAppSoapOut" part="ServerVersion" use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="UninstallApp">
<wsdl:input>
<soap:header message="tns:UninstallAppSoapIn" part="RequestVersion" use="literal"/>
<soap:body use="literal" parts="request"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal" parts="UninstallAppResult"/>
<soap:header message="tns:UninstallAppSoapOut" part="ServerVersion" use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>

<wsdl:message name="DisableAppSoapIn">
<wsdl:part name="request" element="tns:DisableApp"/>
<wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
<wsdl:message name="DisableAppSoapOut"/>
7 Appendix B: Full XML Schema

<table>
<thead>
<tr>
<th>Schema name</th>
<th>Prefix</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://schemas.microsoft.com/exchange/services/2006/types">http://schemas.microsoft.com/exchange/services/2006/types</a></td>
<td>t</td>
<td>7.2</td>
</tr>
<tr>
<td><a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a></td>
<td>tns</td>
<td>7.1</td>
</tr>
</tbody>
</table>

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

7.1 http://schemas.microsoft.com/exchange/services/2006/messages Schema

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

MS-OXWSCEXT-messages.xsd includes the files listed in the following table. For the schema file to operate correctly, these files have 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>
<tr>
<td>MS-OXWSCEXT-types.xsd</td>
<td>section 7.2</td>
</tr>
</tbody>
</table>

```xml
<?xml version="1.0" encoding="UTF-8"?>
           xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
           elementFormDefault="qualified"
           targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
           version="Exchange2016"
           id="messages"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
  <xs:include schemaLocation="MS-OXWSCEXT-messages.xsd"/>
  <xs:import schemaLocation="MS-OXWSCEXT-types.xsd"
              namespace="http://schemas.microsoft.com/exchange/services/2006/types"/>
  <xs:complexType name="ArrayOfAppManifestsType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Manifest"
                    type="xs:base64Binary"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="GetClientAccessToken" type="tns:GetClientAccessTokenType"/>
  <xs:complexType name="GetClientAccessTokenResponseMessageType">
    <xs:complexContent mixed="false">
      <xs:extension base="tns:GetClientAccessTokenResponseMessageType">
        <xs:sequence>
          <xs:element minOccurs="0" name="Token" type="t:ClientAccessTokenType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="GetClientAccessTokenResponse" type="tns:GetClientAccessTokenResponseType"/>
</xs:schema>
```
<xs:complexContent mixed="false">
  <xs:extension base="tns:BaseRequestType">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="ManifestRequests" type="t:NonEmptyArrayOfClientAccessTokenRequestsType"/>
    </xs:sequence>
  </xs:extension>
</xs:complexType>

<xs:element name="GetAppManifestsResponse" type="tns:GetAppManifestsResponseType"/>
<xs:complexType name="GetAppManifestsResponseType">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ResponseMessageType">
      <xs:sequence>
        <xs:element maxOccurs="1" name="Manifests" type="tns:ArrayOfAppManifestsType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="GetAppManifests" type="tns:GetAppManifestsType"/>
<xs:complexType name="GetAppManifestsType">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestType"/>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="GetAppMarketplaceUrlType">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestType"/>
  </xs:complexContent>
</xs:complexType>

<xs:element name="GetAppMarketplaceUrl" type="tns:GetAppMarketplaceUrlType"/>
<xs:complexType name="GetAppMarketplaceUrlResponseMessageType">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ResponseMessageType">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="AppMarketplaceUrl" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="GetAppMarketplaceUrlResponse" type="tns:GetAppMarketplaceUrlResponseMessageType"/>
<xs:complexType name="InstallAppType">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="Manifest" type="xs:base64Binary"/>
        <xs:element minOccurs="0" maxOccurs="1" name="MarketplaceAssetId" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="MarketplaceContentMarket" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="SendWelcomeEmail" type="xs:boolean"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="InstallApp" type="tns:InstallAppType"/>
<xs:complexType name="InstallAppResponseType">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ResponseMessageType">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="WasFirstInstall" type="xs:boolean"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="InstallAppResponse" type="tns:InstallAppResponseType"/>
<xs:complexType name="UninstallAppType">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestType">
      <xs:sequence>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="UninstallAppResponse" type="tns:UninstallAppResponseType"/>
<xs:extension base="tns:BaseRequestType">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="ID" type="xs:string"/>
  </xs:sequence>
</xs:extension>
</xs:complexType>
</xs:element>
<xs:complexType name="UninstallAppResponseType">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
<xs:element name="UninstallAppResponse" type="tns:UninstallAppResponseType"/>
<xs:complexType name="DisableAppType">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:BaseRequestType">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="ID" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="DisableReason" type="t:DisableReasonType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="DisableApp" type="tns:DisableAppType"/>
<xs:complexType name="DisableAppResponseType">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:ResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
<xs:element name="DisableAppResponse" type="tns:DisableAppResponseType"/>
</xs:schema>

7.2 http://schemas.microsoft.com/exchange/services/2006/types Schema

This section contains the contents of the MS-OXWSCEXT-types.xsd file.

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ClientAccessTokenRequestType">
    <xs:sequence>
      <xs:element name="Id" type="xs:string"/>
      <xs:element name="TokenType" type="t:ClientAccessTokenTypeType"/>
      <xs:element minOccurs="0" maxOccurs="1" name="Scope" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ClientAccessTokenType">
    <xs:sequence>
      <xs:element name="Id" type="xs:string"/>
      <xs:element name="TokenType" type="t:ClientAccessTokenTypeType"/>
      <xs:element name="TokenValue" type="xs:string"/>
      <xs:element name="TTL" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ClientAccessTokenTypeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="CallerIdentity"/>
      <xs:enumeration value="ExtensionCallback"/>
      <xs:enumeration value="ScopedToken"/>
      <xs:enumeration value="ExtensionRestApiCallback"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
<xs:simpleType name="DisableReasonType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NoReason"/>
    <xs:enumeration value="OutlookClientPerformance"/>
    <xs:enumeration value="OWAClientPerformance"/>
    <xs:enumeration value="MobileClientPerformance"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="NonEmptyArrayOfClientAccessTokenRequestsType">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" name="TokenRequest" type="t:ClientAccessTokenRequestType"/>
  </xs:sequence>
</xs:complexType>
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 2013
- Microsoft Exchange Server 2016
- Microsoft Exchange Server 2019
- Microsoft Outlook 2013
- Microsoft Outlook 2016
- Microsoft Outlook 2019
- Microsoft Outlook 2021

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.

<1> Section 3.1.4.4.3.3: Exchange 2013 and Outlook 2013 do not support the Scope element.

<2> Section 3.1.4.4.4.1: Exchange 2013 and Outlook 2013 do not support the ScopedToken value.

<3> Section 3.1.4.5.3.1: Exchange 2013 and Outlook 2013 do not support the MarketplaceAssetId element.

<4> Section 3.1.4.5.3.1: Exchange 2013 and Outlook 2013 do not support the MarketplaceContentMarket element.

<5> Section 3.1.4.5.3.1: Exchange 2013 and Outlook 2013 do not support the SendWelcomeEmail element.

<6> Section 3.1.4.5.3.2: Exchange 2013 and Outlook 2013 do not support the WasFirstInstall element.
9 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Revision class</th>
</tr>
</thead>
<tbody>
<tr>
<td>§ Appendix C: Product Behavior</td>
<td>Updated list of supported products.</td>
<td>major</td>
</tr>
</tbody>
</table>
10 Index

A
Abstract data model
  server 12
Applicability 9
Attribute groups 11
Attributes 11

C
Capability negotiation 9
Change tracking 53
Complex types 11

D
Data model - abstract
  server 12
Disabling a mail app example 42

E
Events
  local - server 38
  timer - server 38
Examples
  disabling a mail app 42
  getting a client access token 39
  getting an app Marketplace URL 40
  getting app manifest files 39
  installing a mail app 41
  uninstalling a mail app 43

F
Fields - vendor-extensible 9
Full WSDL 45
Full XML schema 48

  http://schemas.microsoft.com/exchange/service
ts/2006/messages Schema 48

  http://schemas.microsoft.com/exchange/service
ts/2006/types Schema 50

G
Getting a client access token example 39
Getting an app Marketplace URL example 40
Getting app manifest files example 39
Glossary 6
Groups 11

I
Implementer - security considerations 44
Index of security parameters 44
Informative references 8
Initialization
  server 12
  Installing a mail app example 41

Introduction 6

L
Local events
  server 38

M
Message processing
  server 12
Messages
  attribute groups 11
  complex types 11
  elements 10
  enumerated 10
  groups 11
  namespaces 10
  simple types 11
  syntax 10
  transport 10

N
Namespaces 10
Normative references 7

O
Operations
  DisableApp 12
  GetAppManifests 17
  GetAppMarketplaceUrl 21
  GetClientAccessToken 24
  InstallApp 30
  UninstallApp 34
Overview (synopsis) 8

P
Parameters - security index 44
Preconditions 9
Prerequisites 9
Product behavior 52
Protocol Details
  overview 12

R
References 7
  informative 8
  normative 7
Relationship to other protocols 8

S
Security
  implementer considerations 44
  parameter index 44
Sequencing rules