

[MS-OXWSLVID]: Federated Internet Authentication Web Service Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplq@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

| Date | Revision History | Revision Class | Comments |
|------------|------------------|----------------|--------------------------------|
| 11/04/2009 | 1.0.0 | Major | Initial availability |
| 02/10/2010 | 1.1.0 | Minor | Updated the technical content. |
| 05/05/2010 | 1.2.0 | Minor | Updated the technical content. |

Table of Contents

| | |
|--|-----------|
| 1 Introduction | 6 |
| 1.1 Glossary | 6 |
| 1.2 References | 6 |
| 1.2.1 Normative References | 6 |
| 1.2.2 Informative References | 7 |
| 1.3 Overview | 8 |
| 1.4 Relationship to Other Protocols | 8 |
| 1.5 Prerequisites/Preconditions | 8 |
| 1.6 Applicability Statement | 8 |
| 1.7 Versioning and Capability Negotiation | 8 |
| 1.8 Vendor-Extensible Fields | 8 |
| 1.9 Standards Assignments | 8 |
| 2 Messages | 9 |
| 2.1 Transport | 9 |
| 2.2 Common Message Syntax | 9 |
| 2.2.1 Namespaces | 9 |
| 2.2.2 Simple Types | 9 |
| 2.2.3 Complex Types | 9 |
| 2.2.3.1 tns:ArrayOfPropertyType Complex Type | 10 |
| 2.2.3.2 tns:Property Complex Type | 10 |
| 2.2.4 Elements | 10 |
| 2.2.5 Attributes | 11 |
| 2.2.6 Groups | 11 |
| 2.2.7 Attribute Groups | 11 |
| 2.2.8 Messages | 11 |
| 3 Protocol Details | 12 |
| 3.1 ManageDelegationSoap Client 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 | 12 |
| 3.1.4.1 AddUri | 13 |
| 3.1.4.1.1 Elements | 13 |
| 3.1.4.1.1.1 AddUri Element | 13 |
| 3.1.4.1.1.2 AddUriResponse Element | 14 |
| 3.1.4.1.2 Messages | 14 |
| 3.1.4.1.2.1 tns:AddUriSoapIn Message | 14 |
| 3.1.4.1.2.2 tns:AddUriSoapOut Message | 14 |
| 3.1.4.2 CreateAppId | 15 |
| 3.1.4.2.1 Complex Types | 15 |
| 3.1.4.2.1.1 tns:AppIdInfo Complex Type | 15 |
| 3.1.4.2.2 Elements | 16 |
| 3.1.4.2.2.1 CreateAppId Element | 16 |
| 3.1.4.2.2.2 CreateAppIdResponse Element | 16 |
| 3.1.4.2.3 Messages | 17 |
| 3.1.4.2.3.1 tns:CreateAppIdSoapIn Message | 17 |
| 3.1.4.2.3.2 tns:CreateAppIdSoapOut Message | 17 |
| 3.1.4.3 GetDomainInfo | 17 |

| | | |
|-------------|---|----|
| 3.1.4.3.1 | Simple Types | 17 |
| 3.1.4.3.1.1 | tns:DomainState Simple Type | 18 |
| 3.1.4.3.2 | Complex Types | 18 |
| 3.1.4.3.2.1 | tns:DomainInfo Complex Type | 18 |
| 3.1.4.3.3 | Elements | 19 |
| 3.1.4.3.3.1 | GetDomainInfo Element | 19 |
| 3.1.4.3.3.2 | GetDomainInfoResponse Element | 19 |
| 3.1.4.3.4 | Messages | 20 |
| 3.1.4.3.4.1 | tns:GetDomainInfoSoapIn Message | 20 |
| 3.1.4.3.4.2 | tns:GetDomainInfoSoapOut Message | 20 |
| 3.1.4.4 | ReleaseDomain | 20 |
| 3.1.4.4.1 | Elements | 21 |
| 3.1.4.4.1.1 | ReleaseDomain Element | 21 |
| 3.1.4.4.1.2 | ReleaseDomainResponse Element | 21 |
| 3.1.4.4.2 | Messages | 22 |
| 3.1.4.4.2.1 | tns:ReleaseDomainSoapIn Message | 22 |
| 3.1.4.4.2.2 | tns:ReleaseDomainSoapOut Message | 22 |
| 3.1.4.5 | RemoveUri | 22 |
| 3.1.4.5.1 | Elements | 22 |
| 3.1.4.5.1.1 | RemoveUri Element | 22 |
| 3.1.4.5.1.2 | RemoveUriResponse Element | 23 |
| 3.1.4.5.2 | Messages | 23 |
| 3.1.4.5.2.1 | tns:RemoveUriSoapIn Message | 23 |
| 3.1.4.5.2.2 | tns:RemoveUriSoapOut Message | 23 |
| 3.1.4.6 | ReserveDomain | 24 |
| 3.1.4.6.1 | Elements | 24 |
| 3.1.4.6.1.1 | ReserveDomain Element | 24 |
| 3.1.4.6.1.2 | ReserveDomainResponse Element | 25 |
| 3.1.4.6.2 | Messages | 25 |
| 3.1.4.6.2.1 | tns:ReserveDomainSoapIn Message | 25 |
| 3.1.4.6.2.2 | tns:ReserveDomainSoapOut Message | 25 |
| 3.1.4.7 | UpdateAppIdCertificate | 25 |
| 3.1.4.7.1 | Elements | 26 |
| 3.1.4.7.1.1 | UpdateAppIdCertificate Element | 26 |
| 3.1.4.7.1.2 | UpdateAppIdCertificateResponse Element | 27 |
| 3.1.4.7.2 | Messages | 27 |
| 3.1.4.7.2.1 | tns:UpdateAppIdCertificateSoapIn Message | 27 |
| 3.1.4.7.2.2 | tns:UpdateAppIdCertificateSoapOut Message | 27 |
| 3.1.4.8 | UpdateAppIdProperties | 27 |
| 3.1.4.8.1 | Elements | 28 |
| 3.1.4.8.1.1 | UpdateAppIdProperties Element | 28 |
| 3.1.4.8.1.2 | UpdateAppIdPropertiesResponse Element | 29 |
| 3.1.4.8.2 | Messages | 29 |
| 3.1.4.8.2.1 | tns:UpdateAppIdPropertiesSoapIn Message | 29 |
| 3.1.4.8.2.2 | tns:UpdateAppIdPropertiesSoapOut Message | 29 |
| 3.1.5 | Timer Events | 29 |
| 3.1.6 | Other Local Events | 29 |
| 3.2 | Federation Metadata Client Details | 29 |
| 3.2.1 | Abstract Data Model | 29 |
| 3.2.2 | Timers | 30 |
| 3.2.3 | Initialization | 30 |
| 3.2.4 | Message Processing Events and Sequencing | 30 |
| 3.2.5 | Timer Events | 30 |

| | |
|--|-----------|
| 3.2.6 Other Local Events | 30 |
| 4 Protocol Examples..... | 31 |
| 4.1 Registering with a Secure Token Service..... | 31 |
| 4.1.1 Creating an Application Identifier | 31 |
| 4.1.2 Reserving a Federated Organization Domain..... | 32 |
| 4.1.3 Retrieving Domain Information | 33 |
| 4.1.4 Registering a Domain Name | 34 |
| 4.1.5 Removing a Registered Domain Name | 35 |
| 4.1.6 Updating a Certificate | 36 |
| 4.2 Authentication Tokens | 37 |
| 4.2.1 Token Request and Response | 37 |
| 4.2.2 Encrypted and Unencrypted Tokens | 44 |
| 5 Security..... | 49 |
| 5.1 Security Considerations for Implementers..... | 49 |
| 5.2 Index of Security Parameters | 49 |
| 6 Appendix A: Full WSDL..... | 50 |
| 7 Appendix B: Product Behavior..... | 58 |
| 8 Change Tracking..... | 60 |
| 9 Index | 62 |

1 Introduction

This document specifies the Federated Internet Authentication Web Service protocol, which defines the interaction between the server and standard Internet authentication protocols. This document specifies how the server calls external Web services to obtain security tokens that are then used by other Web service protocols to authenticate a transaction.

1.1 Glossary

The following terms are defined in [\[MS-OXGLOS\]](#):

- SOAP body**
- SOAP fault**
- SOAP header**
- SOAP message**
- Uniform Resource Identifier (URI)**
- Uniform Resource Locator (URL)**
- Web Services Description Language (WSDL)**
- WSDL message**
- WSDL port type**
- XML**
- XML namespace**
- XML schema**

The following terms are specific to this document:

Secure Token Service (STS): A Web service that negotiates trust between client applications and services and that provides signed security tokens that can be used for authentication.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

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. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MS-OXGLOS] Microsoft Corporation, "[Exchange Server Protocols Master Glossary](#)", April 2008.

[MS-OXWSMSHR] Microsoft Corporation, "[Folder Sharing Web Service Protocol Specification](#)", November 2009.

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

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, <http://www.ietf.org/rfc/rfc2396.txt>

- [RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <http://www.ietf.org/rfc/rfc2818.txt>
- [RFC3066] Alvestrand, H., "Tags for the Identification of Languages", RFC 3066, BCP 47, January 2001, <http://www.ietf.org/rfc/rfc3066.txt>
- [SAML] Hallam-Baker, P., Kaler, C., Monzillo, R., and Nadalin, A., Eds., "Web Services Security: SAML Token Profile," Oasis Standard, December 2004, <http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0.pdf>
- [SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [WSADDRBIND] Gudgin, M., Hadley, M., Rogers, T., Eds., "Web Services Addressing 1.0 – SOAP Binding", W3C Recommendation, May 2006, <http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/>
- [WSADDRCORE] Gudgin, M., Hadley, M., Rogers, T., Eds., "Web Services Addressing 1.0 – Core", W3C Recommendation, May 2006, <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>
- [WSFED] Lockhart, H., Andersen, S., Bohren, J., et al., "Web Services Federation Language (WS-Federation)", Version 1.1, December 2006, <http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-fed/WS-Federation-V1-1B.pdf>
- [WSECURITY] Nadalin, A., Kaler, C., Monzillo, R., Hallam-Baker, P., Eds., "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)", OASIS Standard Specification, February 2006, <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>
- [WSTRUST] Nadalin, A., Goodner, M., Gudgin, M., Eds., et al., "WS-Trust 1.4", OASIS Standard, February 2009, <http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.doc>
- [XMLDSIG] Eastlake, D. Ed., Reagle, J. Ed., Solo, D. Ed., et al., "XML Signature Syntax and Processing (Second Edition)", W3C Recommendation, June 2008, <http://www.w3.org/TR/2008/REC-xmlsig-core-20080610/>
- [XMLNS] Bray, T., Hollander, D., Layman, A., Eds., et al., "Namespaces in XML 1.0 (Third Edition)", December 2009, <http://www.w3.org/TR/REC-xml-names/>
- [XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>
- [XMLSCHEMA2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>
- [XPATH] Clark, J., and DeRose, S., Eds., "XML Path Language (XPath) Version 1.0", W3C Recommendation, November 1999, <http://www.w3.org/TR/xpath>

1.2.2 Informative References

None.

1.3 Overview

This document specifies the Federated Internet Authentication Web Service protocol, which defines the interactions between the server and standard Internet authentication protocols to provide authentication information to other services on the server. This specification describes how the server uses the following:

- The Managed Delegation Web service, to establish a relationship with a **Secure Token Service (STS)**. The operations that are exposed by the Managed Delegation Web service are specified in section [3.1](#).
- The Federation element, as specified in [\[WSFED\]](#), to provide the security tokens and endpoints that are used to create authentication tokens that can be used to authenticate users and services with other organizations.
- The authentication token that is returned by an STS, as specified in [\[WSTRUST\]](#).

1.4 Relationship to Other Protocols

The Federated Internet Authentication Web Service protocol uses the standard Web Service Federation Language protocol, as specified in [\[WSFED\]](#), and the WS-Trust 1.4 protocol, as specified in [\[WSTRUST\]](#), to provide authentication services for a server. The Folder Sharing Web Service protocol, as specified in [\[MS-OXWSMSHR\]](#), uses this protocol for authentication services.

1.5 Prerequisites/Preconditions

The Federated Internet Authentication Web Service protocol uses services that are provided by external Web services to establish federated relationships between organizations. In order to operate, the protocol requires that the service provide the following:

- The **URL** of a service that provides a Federation Metadata Document, as specified in [\[WSFED\]](#) section 3.1, with the fields and values as specified in section [3.2.1.<1>](#)
- The URL of a delegation management service that provides services, as specified in section [3.1<2>](#).

1.6 Applicability Statement

This protocol is applicable to applications that request federated authentication information on behalf of a client, and applications that expose Web services that provide federated authentication information to servers.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Protocol servers MUST support SOAP over HTTPS. Protocol messages MUST be formatted as specified in [\[SOAP1.1\]](#) or in [\[SOAP1.2/1\]](#). Security tokens MUST be used as specified in [\[WSSECURITY\]](#). Security tokens MUST be exchanged as specified in [\[WSTRUST\]](#). Web service addresses MUST be bound as specified in [\[WSADDRBIND\]](#).

2.2 Common Message Syntax

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

2.2.1 Namespaces

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

| Prefix | Namespace URI | Reference |
|--------|---|---|
| fed | http://schemas.xmlsoap.org/ws/2006/12/federation | [WSFED] |
| wsse | http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd | [WSSECURITY] , Appendix B |
| ds | http://www.w3.org/2000/09/xmldsig# | [XMLDSIG] |
| wsu | http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd | [WSSECURITY] , Appendix A |
| wsa | http://www.w3.org/2005/08/addressing | [WSADDRCORE] , [WSADDRBIND] |
| s | http://www.w3.org/2001/XMLSchema | [XMLNS] |

2.2.2 Simple Types

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

2.2.3 Complex Types

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

| Complex Type | Description |
|---------------------------------|--|
| ArrayOfProperty | Specifies an array of property name/value pairs for a managed delegate relationship. |
| Property | Specifies a name value pair for a managed delegate relationship. |

2.2.3.1 tns:ArrayOfPropertyType Complex Type

The **ArrayOfPropertyType** complex type specifies one or more **Property** complex type (section [2.2.3.2](#)) name/value pairs.

```
<xs:complexType name="ArrayOfPropertyType">
  <xs:sequence>
    <xs:element name="Property"
      type="tns:Property"
      minOccurs="0"
      maxOccurs="unbounded"
    />
  </xs:sequence>
</xs:complexType>
```

Child Elements

| Element | Type | Description |
|----------|------------------------------|--|
| Property | tns:Property | A name/value pair that describes a managed delegation relationship property. |

2.2.3.2 tns:Property Complex Type

The **Property** complex type specifies a managed delegation property as a name/value pair.

```
<xs:complexType name="Property">
  <xs:sequence>
    <xs:element name="Name"
      type="s:string"
      maxOccurs="1"
      minOccurs="0"
    />
    <xs:element name="Value"
      type="s:string"
      maxOccurs="1"
      minOccurs="0"
    />
  </xs:sequence>
</xs:complexType>
```

Child Elements

| Element | Type | Description |
|---------|----------|--|
| Name | s:string | Specifies the name of the property. |
| Value | s:string | Specifies the value of the property expressed as a string. |

2.2.4 Elements

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

2.2.5 Attributes

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

2.2.6 Groups

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

2.2.7 Attribute Groups

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

2.2.8 Messages

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

3 Protocol Details

The Federated Internet Authentication Web service protocol does not act as a server, and does not expose any services to outside callers. This specification describes the server's interactions as a client to external services.

3.1 ManageDelegationSoap Client Details

The Federated Internet Authentication Web Service protocol uses the following operations that are exposed by the ManageDelegationSoap Web service.

| Operation | Description |
|--|---|
| AddUri (section 3.1.4.1) | Registers a URI with the federation management service. |
| CreateAppId (section 3.1.4.2) | Creates an application identifier for an organization with the federation management service. |
| GetDomainInfo (section 3.1.4.3) | Gets domain status information from the federation management service. |
| ReleaseDomain (section 3.1.4.4) | Removes a domain from the federation management service. |
| RemoveUri (section 3.1.4.5) | Removes a registered URI from the federation management service. |
| ReserveDomain (section 3.1.4.6) | Verifies that a domain is to be managed by the specified application identifier. |
| UpdateAppIdCertificate (section 3.1.4.7) | Updates the security certificate that is associated with an application identifier. |
| UpdateAppIdProperties (section 3.1.4.8) | Updates the organizational information that is associated with an application identifier. |

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

This protocol uses the operations that are listed in the following table.

| Operation | Description |
|---|--|
| AddUri (section 3.1.4.1) | Registers a URI with the federation management service. |
| CreateAppId (section 3.1.4.2) | Creates an application identifier for an organization with the |

| Operation | Description |
|--|---|
| | federation management service. |
| GetDomainInfo (section 3.1.4.3) | Gets domain status information from the federation management service. |
| ReleaseDomain (section 3.1.4.4) | Removes a domain from the federation management service. |
| RemoveUri (section 3.1.4.5) | Removes a registered URI from the federation management service. |
| ReserveDomain (section 3.1.4.6) | Verifies that a domain should be managed by the specified application identifier. |
| UpdateAppIdCertificate (section 3.1.4.7) | Updates the security certificate associated with an application identifier. |
| UpdateAppIdProperties (section 3.1.4.8) | Updates the organizational information associated with an application identifier. |

3.1.4.1 AddUri

The **AddUri** operation registers the URL of an organization participating in the federation management service.

```
<wsdl:operation name="AddUri">
  <wsdl:input message="tns:AddUriSoapIn" />
  <wsdl:output message="tns:AddUriSoapOut" />
</wsdl:operation>
```

Request

| Message Format | Description |
|----------------------------------|--|
| tns:AddUriSoapIn | Specifies the SOAP message that requests the registration of a URI. |

Response

| Message Format | Description |
|-----------------------------------|--|
| tns:AddUriSoapOut | Specifies the SOAP message that is returned by the server in response. |

3.1.4.1.1 Elements

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

3.1.4.1.1.1 AddUri Element

The `<AddUri>` element specifies the URI that is to be added to the federation management service by the **AddUri** operation (section [3.1.4.1](#)).

```
<xs:element name="AddUri">
  <xs:complexType>
```

```

<xs:sequence>
  <xs:element name="ownerAppId"
    type="s:string"
    maxOccurs="1"
    minOccurs="0"
  />
  <xs:element name="uri"
    type="s:string"
    maxOccurs="1"
    minOccurs="0"
  />
</xs:sequence>
</xs:complexType>
</xs:element>

```

Child Elements

| Element | Type | Description |
|------------|----------|---|
| ownerAppId | s:string | Specifies the application identifier that is assigned to the entity requesting that the URI be registered with a federation management service. |
| uri | s:string | Specifies the URI to register with the federation management service. |

3.1.4.1.1.2 AddUriResponse Element

The <AddUriResponse> element specifies the response from the **AddUri** operation (section [3.1.4.1.1](#)).

```

<xs:element name="AddUriResponse">
  <xs:complexType />
</xs:element>

```

3.1.4.1.2 Messages

The following **WSDL message** definitions are specific to this operation.

3.1.4.1.2.1 tns:AddUriSoapIn Message

The **AddUriSoapIn** message contains one part.

| Part Name | Element/Type | Description |
|------------|----------------------------|---|
| parameters | tns:AddUri | Specifies the request to register a URI with the federation management service. |

3.1.4.1.2.2 tns:AddUriSoapOut Message

The **AddUriSoapOut** message contains one part.

| Part name | Element/Type | Description |
|------------|--|-------------------------|
| parameters | tns:AddUriResponse Element | Specifies the response. |

3.1.4.2 CreateAppId

The **CreateAppId** operation creates an identifier for an organization that participates in a federation management service. The identifier that is returned by the **CreateAppId** operation is used when calling operations on the federation management server to identify the organization that is making the request.

```
<wsdl:operation name="CreateAppId">
  <wsdl:input message="tns:CreateAppIdSoapIn" />
  <wsdl:output message="tns:CreateAppIdSoapOut" />
</wsdl:operation>
```

Request

| Message Format | Description |
|---------------------------------------|--|
| tns:CreateAppIdSoapIn | Specifies the SOAP message that requests the application identifier. |

Response

| Message Format | Description |
|--|--|
| tns:CreateAppIdSoapOut | Specifies the SOAP message that is returned by the server in response. |

The **CreateAppID** operation requires that the certificate specified in the input message be attached as a **SOAP header** to the request.

3.1.4.2.1 Complex Types

The following XML schema complex types are specific to this operation.

3.1.4.2.1.1 tns:AppIdInfo Complex Type

The **AppIdInfo** complex type specifies an application identifier and the associated administrative key.

```
<xs:complexType name="AppIdInfo">
  <xs:sequence>
    <xs:element name="AppId"
      type="s:string"
    />
    <xs:element name="AdminKey"
      type="s:string"
    />
  </xs:sequence>
</xs:complexType>
```

Child Elements

| Element | Type | Description |
|----------|----------|--|
| AppId | s:string | Specifies an application identifier. |
| AdminKey | s:string | Specifies the administrative key that is associated with the application identifier. |

3.1.4.2.2 Elements

The following XML schema elements are specific to this operation.

3.1.4.2.2.1 CreateAppId Element

The <CreateAppId> element specifies the information that is required to establish a relationship with a federation management service.

```
<xs:element name="CreateAppId">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="cerfificate"
        type="s:string"
        minOccurs="0"
        maxOccurs="1"
      />
      <xs:element name="properties"
        type="tns:ArrayOfProperty"
      />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Child Elements

| Element | Type | Description |
|-------------|-------------------------------------|--|
| cerfificate | s:string | Specifies the certificate that will be used for application identifier management and for encryption of the delegation ticket for this domain. MUST be a base-64 encoded string. |
| properties | tns:ArrayOfProperty | Specifies additional information about the organization. Can be present. |

3.1.4.2.2.2 CreateAppIdResponse Element

The <CreateAppIdResponse> element specifies the response from the **CreateAppId** operation (section [3.1.4.2](#)) that contains an application identifier and administrative key.

```
<xs:element name="CreateAppIdResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CreateAppIdResult"
        type="tns:AppIdInfo"
      />
    </xs:sequence>
  </xs:complexType>
```


</xs:element>

Child Elements

| Element | Type | Description |
|-------------------|-------------------------------|--|
| CreateAppIdResult | tns:AppIdInfo | Specifies an application identifier and the associated administrative key. |

3.1.4.2.3 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.2.3.1 tns:CreateAppIdSoapIn Message

The **CreateAppIdSoapIn** message contains one part.

| Part Name | Element/Type | Description |
|------------|------------------------|---|
| parameters | tns:CreateAppId | Contains the request to create an application identifier. |

3.1.4.2.3.2 tns:CreateAppIdSoapOut Message

The **CreateAppIdSoapOut** message contains one part.

| Part Name | Element/Type | Description |
|------------|--------------------------------|---|
| parameters | tns:CreateAppIdResponse | Specifies the response that contains the application identifier and administrative key. |

3.1.4.3 GetDomainInfo

The **GetDomainInfo** operation retrieves federation status information for a domain.

Request

| Message Format | Description |
|---|---|
| tns:GetDomainInfoSoapIn | Specifies the SOAP message that requests domain status information. |

Response

| Message Format | Description |
|--|--|
| tns:GetDomainInfoSoapOut | Specifies the SOAP message that is returned by the server in response. |

3.1.4.3.1 Simple Types

The following XML schema simple type definitions are specific to this operation.

3.1.4.3.1.1 tns:DomainState Simple Type

The **DomainState** simple type specifies the possible states that can be returned by the **GetDomainInfo** operation (section [3.1.4.3](#)).

```
<xs:simpleType name="DomainState">
  <xs:restriction
    base="s:string"
  >
    <xs:enumeration
      value="PendingActivation"
    />
    <xs:enumeration
      value="Active"
    />
    <xs:enumeration
      value="PendingRelease"
    />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

The following values are defined by the **DomainState** simple type:

| Value | Description |
|-------------------|--|
| PendingActivation | The request to create a domain has been received but it is not yet active. |
| Active | The domain is active. |
| PendingRelease | The request to release a domain has been received, but the domain has not yet been released. |

3.1.4.3.2 Complex Types

The following XML schema complex types are specific to this operation.

3.1.4.3.2.1 tns:DomainInfo Complex Type

The **DomainInfo** complex type defines the domain information that is returned by the **GetDomainInfo** operation (section [3.1.4.3](#)).

```
<xs:complexType name="DomainInfo">
  <xs:sequence>
    <xs:element name="DomainName"
      type="s:string"
      maxOccurs="1"
      minOccurs="0"
    />
    <xs:element name="AppId"
      type="s:string"
      maxOccurs="1"
      minOccurs="0"
    />
  </xs:sequence>
</xs:complexType>
```

```

<xs:element name="DomainState"
  type="tns:DomainState"
  maxOccurs="1"
  minOccurs="1"
/>
</xs:sequence>
</xs:complexType>

```

Child Elements

| Element | Type | Description |
|-------------|---------------------------------|--|
| DomainName | s:string | Specifies the registered name of the domain. |
| AppId | s:string | Specifies the application identifier that is associated with the domain. |
| DomainState | tns:DomainState | Specifies the current state of the domain. MUST be present. |

3.1.4.3.3 Elements

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

3.1.4.3.3.1 GetDomainInfo Element

The **GetDomainInfo** element specifies the information that is needed to request the current status of a domain.

```

<xs:element name="GetDomainInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ownerAppId"
        type="s:string"
      />
      <xs:element name="domainName"
        type="s:string"
      />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Child Elements

| Element | Type | Description |
|------------|----------|---|
| ownerAppId | s:string | Specifies the application identifier of the domain owner. |
| domainName | s:string | Specifies the domain for which information is to be returned. |

3.1.4.3.3.2 GetDomainInfoResponse Element

The **GetDomainInfoResponse** element specifies the response from a **GetDomainInfo** operation (section [3.1.4.3](#)) request.

```

<xs:element name="GetDomainInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="GetDomainInfoResult"
        type="tns:DomainState"
        minOccurs="0"
        maxOccurs="1"
      />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Child Elements

| Element | Type | Description |
|---------------------|---------------------------------|--|
| GetDomainInfoResult | tns:DomainState | Specifies the domain status information. |

3.1.4.3.4 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.3.4.1 tns:GetDomainInfoSoapIn Message

The **GetDomainInfoSoapIn** message defines one part.

| Part name | Element/type | Description |
|------------|-----------------------------------|------------------------|
| parameters | tns:GetDomainInfo | Specifies the request. |

3.1.4.3.4.2 tns:GetDomainInfoSoapOut Message

The **GetDomainInfoSoapOut** message defines one part.

| Part name | Element/type | Description |
|------------|---|-------------------------|
| parameters | tns:GetDomainInfoResponse | Specifies the response. |

3.1.4.4 ReleaseDomain

The **ReleaseDomain** operation releases the specified domain from federation management services.

```

<wsdl:operation name="ReleaseDomain">
  <wsdl:input message="tns:ReleaseDomainSoapIn" />
  <wsdl:output message="tns:ReleaseDomainSoapOut" />
</wsdl:operation>

```

Request

| Message Format | Description |
|---|---|
| tns:ReleaseDomainSoapIn | Specifies the SOAP message that requests that the domain be released. |

Response

| Message Format | Description |
|--|--|
| tns:ReleaseDomainSoapOut | Specifies the SOAP message that is returned by the server in response. |

3.1.4.4.1 Elements

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

3.1.4.4.1.1 ReleaseDomain Element

The <ReleaseDomain> element specifies the information that is required for the **ReleaseDomain** operation (section [3.1.4.4](#)).

```
<xs:element name="ReleaseDomain">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ownerAppId"
        type="s:string"
      />
      <xs:element name="domainName"
        type="s:string"
      />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Child Elements

| Element | Type | Description |
|------------|----------|--|
| ownerAppId | s:string | Specifies the application identifier assigned to the domain manager when the domain was registered with the federation management service. |
| domainName | s:string | Specifies the domain to release. |

3.1.4.4.1.2 ReleaseDomainResponse Element

The <ReleaseDomainResponse> element specifies the response from the **ReleaseDomain** operation (section [3.1.4.4](#)).

```
<xs:element name="ReleaseDomainResponse">
  <xs:complexType />
</xs:element>
```

3.1.4.4.2 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.4.2.1 tns:ReleaseDomainSoapIn Message

The **ReleaseDomainSoapIn** message defines one part.

| PartName | Element/Type | Description |
|------------|---|--|
| parameters | tns:ReleaseDomain Element | Specifies the request to release a domain. |

3.1.4.4.2.2 tns:ReleaseDomainSoapOut Message

The **ReleaseDomainSoapOut** message defines one part.

| Part Name | Element/Type | Description |
|------------|---|--|
| parameters | tns:ReleaseDomainResponse | Defines the response from the operation. |

3.1.4.5 RemoveUri

The **RemoveUri** operation removes a previously registered URI from the federation management service.

```
<wsdl:operation name="RemoveUri">
  <wsdl:input message="tns:RemoveUriSoapIn" />
  <wsdl:output message="tns:RemoveUriSoapOut" />
</wsdl:operation>
```

Request

| Message Format | Description |
|-------------------------------------|--|
| tns:RemoveUriSoapIn | Specifies the SOAP message that requests that a URI be released. |

Response

| Message Format | Description |
|--------------------------------------|--|
| tns:RemoveUriSoapOut | Specifies the SOAP message that is returned by the server in response. |

3.1.4.5.1 Elements

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

3.1.4.5.1.1 RemoveUri Element

The **<RemoveUri>** element specifies the application identifier and URI to remove for the **RemoveUri** operation (section [3.1.4.5](#)).

```
<xs:element name="RemoveUri">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="ownerAppId"
      type="s:string"
      maxOccurs="1"
      minOccurs="0"
    />
    <xs:element name="uri"
      type="s:string"
      maxOccurs="1"
      minOccurs="0"
    />
  </xs:sequence>
</xs:complexType>
</xs:element>

```

Child Elements

| Element | Type | Description |
|------------|----------|--|
| ownerAppId | s:string | Specifies the application identifier of the organization that is removing the URI. |
| uri | s:string | Specifies the URI to remove. |

3.1.4.5.1.2 RemoveUriResponse Element

The <RemoveUriResponse> element specifies the response from the **RemoveUri** operation (section [3.1.4.5](#)).

```

<xs:element name="RemoveUriResponse">
  <xs:complexType />
</xs:element>

```

3.1.4.5.2 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.5.2.1 tns:RemoveUriSoapIn Message

The **RemoveUriSoapIn** message defines one part.

| Part Name | Element/Type | Description |
|------------|-------------------------------|--|
| parameters | tns:RemoveUri | Specifies the application identifier of the URI owner and the URI to remove from the federation management server. |

3.1.4.5.2.2 tns:RemoveUriSoapOut Message

The **RemoveUriSoapOut** message defines one part.

| Part Name | Element/Type | Description |
|------------|---------------------------------------|--|
| parameters | tns:RemoveUriResponse | Specifies the response from the operation. |

3.1.4.6 ReserveDomain

The **ReserveDomain** operation verifies that a specified domain is to be associated with an application identifier.

```
<wsdl:operation name="ReserveDomain">
  <wsdl:input message="tns:ReserveDomainSoapIn" />
  <wsdl:output message="tns:ReserveDomainSoapOut" />
</wsdl:operation>
```

Request

| Message Format | Description |
|---|--|
| tns:ReserveDomainSoapIn | Specifies the SOAP message that requests validation of a domain. |

Response

| Message Format | Description |
|--|--|
| tns:ReserveDomainSoapOut | Specifies the SOAP message that is returned by the server in response. |

3.1.4.6.1 Elements

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

3.1.4.6.1.1 ReserveDomain Element

The <ReserveDomain> element specifies the information that is required to reserve a domain for federation management by using the **ReserveDomain** operation (section [3.1.4.6](#)).

```
<xs:element name="ReserveDomain">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ownerAppId"
        type="s:string"
        minOccurs="0"
        maxOccurs="1"
      />
      <xs:element name="domainName"
        type="s:string"
        maxOccurs="1"
        minOccurs="0"
      />
      <xs:element name="programId"
        type="s:string"
        maxOccurs="1"
        minOccurs="0"
      />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```



```

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

```

Child Elements

| Element | Type | Description |
|------------|----------|--|
| ownerAppId | s:string | Specifies the application identifier of the organization that wants to reserve the domain. |
| domainName | s:string | Specifies the domain name of the domain to reserve for federation management. |
| programId | s:string | Reserved for future use. <3> |

3.1.4.6.1.2 ReserveDomainResponse Element

The `<ReserveDomainResponse>` element specifies the response from the **ReserveDomain** operation (section [3.1.4.6](#)).

```

<xs:element name="ReserveDomainResponse">
  <xs:complexType />
</xs:element>

```

3.1.4.6.2 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.6.2.1 tns:ReserveDomainSoapIn Message

The **ReserveDomainSoapIn** message defines one part.

| Part Name | Element/Type | Description |
|------------|-----------------------------------|--|
| parameters | tns:ReserveDomain | Specifies the request to reserve a domain. |

3.1.4.6.2.2 tns:ReserveDomainSoapOut Message

The **ReserveDomainSoapOut** message defines one part.

| Part name | Element/type | Description |
|------------|---|--|
| parameters | tns:ReserveDomainResponse | Specifies the response from the operation. |

3.1.4.7 UpdateAppIdCertificate

The **UpdateAppIdCertificate** operation updates the security certificate that is associated with an application identifier. After the certificate is updated, all subsequent calls to federation management operations must use the new certificate for identification and encryption.

```

<wsdl:operation name="UpdateAppIdCertificate">
  <soap12:operation
    soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/UpdateAppIdCertificate"
    style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>

```

Request

| Message Format | Description |
|--|---|
| tns:UpdateAppIdCertificateSoapIn | Specifies the SOAP message that requests that a certificate be updated. |

Response

| Message Format | Description |
|---|--|
| tns:UpdateAppIdCertificateSoapOut | Specifies the SOAP message that is returned by the server in response. |

3.1.4.7.1 Elements

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

3.1.4.7.1.1 UpdateAppIdCertificate Element

The <UpdateAppIdCertificate> element specifies the authentication information and new certificate to replace the existing certificate for the **UpdateAppIdCertificate** operation ([3.1.4.7](#)).

```

<xs:element name="UpdateAppIdCertificate"
  maxOccurs="1"
  minOccurs="0"
  >
  <xs:complexType>
    <xs:sequence>
      <xs:element name="appId"
        type="s:string"
        maxOccurs="1"
        minOccurs="0"
        />
      <xs:element name="appIdAdminKey"
        type="s:string"
        />
      <xs:element name="newCertificate"
        type="s:string"
        maxOccurs="1"
        minOccurs="0"
        />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

</xs:complexType>
</xs:element>

```

Child Elements

| Element | Type | Description |
|----------------|----------|---|
| appId | s:string | Specifies the application identifier for the organization that is changing the security certificate that is associated with the application identifier. |
| appIdAdminKey | s:string | Specifies the administrative key that was associated with the application identifier when the application identifier was created. |
| newCertificate | s:string | Specifies the new security certificate as a base-64 encoded string. |

3.1.4.7.1.2 UpdateAppIdCertificateResponse Element

The <UpdateAppIdCertificateResponse> element specifies the response from the **UpdateAppIdCertificate** operation (section [3.1.4.7](#))

```

<xs:element name="UpdateAppIdCertificateResponse">
  <xs:complexType />
</xs:element>

```

3.1.4.7.2 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.7.2.1 tns:UpdateAppIdCertificateSoapIn Message

The **UpdateAppIdCertificateSoapIn** message defines one part.

| Part Name | Element/Type | Description |
|------------|--|---|
| parameters | tns:UpdateAppIdCertificate | Specifies the request to update the security certificate that is associated with an application identifier. |

3.1.4.7.2.2 tns:UpdateAppIdCertificateSoapOut Message

The **UpdateAppIdCertificateSoapOut** message defines one part.

| Part Name | Element/Type | Description |
|------------|--|---|
| parameters | tns:UpdateAppIdCertificateResponse | Specifies the response from the server. |

3.1.4.8 UpdateAppIdProperties

The **UpdateAppIdProperties** operation updates the additional information about an organization that is stored with the federation management service.

```

<wsdl:operation name="UpdateAppIdProperties">
  <wsdl:input message="tns:UpdateAppIdPropertiesSoapIn" />
  <wsdl:output message="tns:UpdateAppIdPropertiesSoapOut" />
</wsdl:operation>

```

Request

| Message Format | Description |
|---|---|
| tns:UpdateAppIdPropertiesSoapIn | Specifies the SOAP message that requests that organization information be modified. |

Response

| Message Format | Description |
|--|--|
| tns:UpdateAppIdPropertiesSoapOut | Specifies the SOAP message that is returned by the server in response. |

3.1.4.8.1 Elements

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

3.1.4.8.1.1 UpdateAppIdProperties Element

The <UpdateAppIdProperties> element specifies the organization properties to modify with the **UpdateAppIdProperties** operation ([3.1.4.8](#)).

```

<xs:element name="UpdateAppIdProperties">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ownerAppId"
        type="s:string"
        maxOccurs="1"
        minOccurs="0"
      />
      <xs:element name="properties"
        type="tns:ArrayOfProperty"
        maxOccurs="1"
        minOccurs="0"
      />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Child Elements

| Element | Type | Description |
|------------|----------|---|
| ownerAppId | s:string | Specifies the application identifier of the organization that is changing properties. |

| Element | Type | Description |
|------------|-------------------------------------|---|
| properties | tns:ArrayOfProperty | Specifies one or more properties to modify. |

3.1.4.8.1.2 UpdateAppIdPropertiesResponse Element

The <UpdateAppIdPropertiesResponse> element specifies the response from the **UpdateAppIdProperties** operation (section [3.1.4.8](#)).

```
<xs:element name="UpdateAppIdPropertiesResponse">
  <xs:complexType />
</xs:element>
```

3.1.4.8.2 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.8.2.1 tns:UpdateAppIdPropertiesSoapIn Message

The **UpdateAppIdPropertiesSoapIn** message specifies one part.

| Part name | Element/type | Description |
|------------|---|-------------------------------------|
| parameters | tns:UpdateAppIdProperties | Specifies the properties to modify. |

3.1.4.8.2.2 tns:UpdateAppIdPropertiesSoapOut Message

The **UpdateAppIdPropertiesSoapOut** message defines one part.

| Part name | Element/type | Description |
|------------|---|-----------------------|
| parameters | tns:UpdateAppIdPropertiesResponse | Defines the response. |

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Federation Metadata Client Details

The Federated Authentication Web service protocol uses elements from the federation metadata **XML** document, as specified in [\[WSFED\]](#). [<4>](#)

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. the described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations

adhere to this model as long as their external behavior is consistent with that described in this document.

The Federation Metadata XML document, as specified in [\[WSFED\]](#), is stateless; however, the server can cache certain values that are contained in the Federation Metadata XML document to improve performance.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Message Processing Events and Sequencing

None.

3.2.5 Timer Events

None.

3.2.6 Other Local Events

None.

4 Protocol Examples

The following examples show the XML messages that are used by the Federated Internet Authentication Web Service protocol. Where the Federated Internet Authentication Web Service protocol requires specific values in an element of the XML document, the element node is described by using the syntax specified in [\[XPATH\]](#).

4.1 Registering with a Secure Token Service

The following examples show the XML messages that are used by the Federated Internet Authentication Web Service protocol to communicate with the Managed Delegation Web service that is exposed by a Secure Token Service. Where the Federated Internet Authentication Web Service protocol requires specific values in an element of the XML document, the element node is described by using the syntax specified in [\[XPATH\]](#).

4.1.1 Creating an Application Identifier

This example shows the request and response messages that are sent to and received from the **CreateAppId** operation (section [3.1.4.2](#)).

Request XML

The following is an example of the request that sent to the **CreateAppId** operation (section [3.1.4.2](#)).

```
<?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:Body>
    <CreateAppId xmlns="http://domains.live.com/Service/ManageDelegation/V1.0">
      <certificate>MIIFCjCCBLSgAwIBAgIKFZsHigAGA...</certificate>
    </CreateAppId>
  </soap:Body>
</soap:Envelope>
```

The following describes the required attributes and elements that are used in the example:

/soap:Envelope/soap:Body/CreateAppId/certificate: The certificate that will be used to identify requests from the organization and to encrypt information sent to the organization. MUST be a base 64-encoded string.

Response XML

The following is an example of the response that is returned by the **CreateAppId** operation (section [3.1.4.2](#)).

```
<?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:Body>
    <CreateAppIdResponse xmlns="http://domains.live.com/Service/ManageDelegation/V1.0">
```

```

    <CreateAppIdResult>
      <AppId>0000000060000EB9</AppId>
      <AdminKey>6MoW1lqVuL/sYZFCNPcGRhn+dyVX4TR4J9xFZsB7jKU=</AdminKey>
    </CreateAppIdResult>
  </CreateAppIdResponse>
</soap:Body>
</soap:Envelope>

```

The following describes the required attributes and elements that are used in the example:

/soap:Envelope/soap:Body/CreateAppIdResponse/CreateAppIdResult/AppId: The application identifier that is assigned to the organization by the STS. The application identifier can be any combination of letters and numbers.

/soap:Envelope/soap:Body/CreateAppIdResponse/CreateAppIdResult/AdminKey: The administrative key that is assigned to the organization by the STS. This key is used to identify the organization when changing administrative information that is maintained by the STS. The administrative key can be any combination of letters and numbers.

4.1.2 Reserving a Federated Organization Domain

This example shows the request and response messages that are sent to and received from the **ReserveDomain** operation (section [3.1.4.6](#)).

Request XML

The following is an example of the request that is sent to the **ReserveDomain** operation (section [3.1.4.6](#)).

```

<?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:Body>
    <ReserveDomain xmlns="http://domains.live.com/Service/ManageDelegation/V1.0">
      <ownerAppId>0000000060000EB9</ownerAppId>
      <domainName>contoso.com</domainName>
      <programId></programId>
    </ReserveDomain>
  </soap:Body>
</soap:Envelope>

```

The following describes the required attributes and elements that are used in the example:

/soap:Envelope/soap:Body/ReserveDomain/ownerAppId: The application identifier that is assigned to the organization by the STS. This value is returned in response to the **CreateAppId** operation (section [3.1.4.2](#)).

/soap:Envelope/soap:Body/ReserveDomain/domainName: The domain name of the organization.

/soap:Envelope/soap:Body/ReserveDomain/programId: This element is reserved for future use. [<5>](#)

Response XML

The following is an example of the response that is returned by the **ReserveDomain** operation (section [3.1.4.6](#)).

```
<?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:Body>
    <ReserveDomainResponse xmlns="http://domains.live.com/Service/ManageDelegation/V1.0" />
  </soap:Body>
</soap:Envelope>
```

4.1.3 Retrieving Domain Information

This example shows the request and response messages that are sent to and received from the **GetDomainInfo** operation (section [3.1.4.3](#)).

Request XML

The following is an example of the request that is sent to the **GetDomainInfo** operation (section [3.1.4.3](#)).

```
<?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:Body>
    <GetDomainInfo xmlns="http://domains.live.com/Service/ManageDelegation/V1.0">
      <ownerAppId>0000000060000EB9</ownerAppId>
      <domainName>contoso.com</domainName>
    </GetDomainInfo>
  </soap:Body>
</soap:Envelope>
```

The following describes the required attributes and elements that are used in the example:

/soap:Envelope/soap:Body/GetDomainInfo/ownerAppId: The application identifier that is assigned to the organization by the STS. The application identifier can be any combination of letters and numbers.

/soap:Envelope/soap:Body/GetDomainInfo/domainName: The domain name of the organization.

Response XML

The following is an example of the response that is returned by the **GetDomainInfo** operation (section [3.1.4.3](#)).

```
<?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:Body>
    <GetDomainInfoResponse xmlns="http://domains.live.com/Service/ManageDelegation/V1.0">
      <GetDomainInfoResult>
        <DomainName>vyotqn-dom.extest.microsoft.com</DomainName>
        <AppId>0000000060000EB9</AppId>
        <DomainState>Active</DomainState>
      </GetDomainInfoResult>
    </GetDomainInfoResponse>
  </soap:Body>
</soap:Envelope>

```

The following describes the required attributes and elements that are used the example:

/soap:Envelope/soap:Body/GetDomainInfoResponse/GetDomainInfoResult/DomainName: The domain registered by the organization with the STS.

/soap:Envelope/soap:Body/GetDomainInfoResponse/GetDomainInfoResult/AppId: The application identifier that is assigned to the organization by the STS. The application identifier can be any combination of letters and numbers.

/soap:Envelope/soap:Body/GetDomainInfoResponse/GetDomainInfoResult/DomainState: The current state of the domain. The possible states are specified by the **DomainState** simple type (section [3.1.4.3.1.1](#)).

4.1.4 Registering a Domain Name

This example shows the request and response messages that are sent to and received from the **AddUri** operation (section [3.1.4.1](#)).

Request XML

The following is an example of the request that is sent to the **AddUri** operation (section [3.1.4.1](#)).

```

<?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:Body>
    <AddUri xmlns="http://domains.live.com/Service/ManageDelegation/V1.0">
      <ownerAppId>0000000060000EB9</ownerAppId>
      <uri>VYOTQN-DOM.EXTEST.MICROSOFT.COM</uri>
    </AddUri>
  </soap:Body>
</soap:Envelope>

```

The following describes the required attributes and elements that are used in the example:

/soap:Envelope/soap:Body/AddUri/ownerAppId: The application identifier that is assigned to the organization by the STS. The application identifier can be any combination of letters and numbers.

/soap:Envelope/soap:Body/AddUri/uri: The domain name of the organization.

Response XML

The following is an example of the response that is returned by the **AddUri** operation (section [3.1.4.1](#)).

```
<?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:Body>
    <AddUriResponse xmlns="http://domains.live.com/Service/ManageDelegation/V1.0" />
  </soap:Body>
</soap:Envelope>
```

4.1.5 Removing a Registered Domain Name

This example shows the request and response messages that are sent to and received from the **RemoveUri** operation (section [3.1.4.5](#)).

Request XML

The following is an example of the request that is sent to the **RemoveUri** operation (section [3.1.4.5](#)).

```
<?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:Body>
    <RemoveUri xmlns="http://domains.live.com/Service/ManageDelegation/V1.0">
      <ownerAppId>0000000060000EB9</ownerAppId>
      <uri>contoso.com</uri>
    </RemoveUri>
  </soap:Body>
</soap:Envelope>
```

The following describes the required attributes and elements that are used in the example:

/soap:Envelope/soap:Body/RemoveUri/ownerAppId: The application identifier that is assigned to the organization by the STS. The application identifier can be any combination of letters and numbers.

/soap:Envelope/soap:Body/RemoveUri/uri: The organization domain name to remove.

Response XML

The following is an example of the response that is returned by the **RemoveUri** operation (section [3.1.4.5](#)).

```

<?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:Body>
    <RemoveUriResponse xmlns="http://domains.live.com/Service/ManageDelegation/V1.0" />
  </soap:Body>
</soap:Envelope>

```

4.1.6 Updating a Certificate

This example shows the request and response messages that are sent to and received from the **UpdateAppIdCertificate** operation (section [3.1.4.7](#)).

Request XML

The following is an example of the request that is sent to the **UpdateAppIdCertificate** operation (section [3.1.4.7](#)).

```

<?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:Body>
    <UpdateAppIdCertificate xmlns="http://domains.live.com/Service/ManageDelegation/V1.0">
      <appId>0000000060000EB9</appId>
      <appIdAdminKey>6MoW1lqVuL/sYZFCNPcGRhn+dyVX4TR4J9xFZsB7jKU=</appIdAdminKey>
      <newCertificate>MIIFTTCBPegAwIBAgIKI1... </newCertificate>
    </UpdateAppIdCertificate>
  </soap:Body>
</soap:Envelope>

```

The following describes the required attributes and elements that are used in the example:

/soap:Envelope/soap:Body/UpdateAppIdCertificate/appId: The application identifier that is assigned to the organization by the STS. The application identifier can be any combination of letters and numbers.

/soap:Envelope/soap:Body/UpdateAppIdCertificate/apIdAdminKey: The administrative key that is assigned to the organization by the STS.

/soap:Envelope/soap:Body/UpdateAppIdCertificate/newCertificate: The new certificate that will be used to identify requests from the organization and to encrypt information that is sent to the organization. MUST be a base 64-encoded string.

Response XML

The following is an example of the response that is returned by the **UpdateAppIdCertificate** operation (section [3.1.4.7](#)).

```

<?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:Body>
    <UpdateAppIdCertificateResponse
xmlns="http://domains.live.com/Service/ManageDelegation/V1.0" />
  </soap:Body>
</soap:Envelope>

```

4.2 Authentication Tokens

The following examples show the request for a token and the response from the Secure Token Service that contains token, and the encrypted and unencrypted tokens.

4.2.1 Token Request and Response

This section shows the token request and response that is sent to and received from the Secure Token Service.

Token Request

The following is an example of the token request that is sent to an STS.

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:t="http://schemas.xmlsoap.org/ws/2005/02/trust"
xmlns:auth="http://schemas.xmlsoap.org/ws/2006/12/authorization"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <s:Header>
    <a:To s:mustUnderstand="1" u:Id="_1">https://login.live-
int.com:44329/liveidSTS.srf</a:To>
    <a:Action
s:mustUnderstand="1">http://schemas.xmlsoap.org/ws/2005/02/trust/RST/Issue</a:Action>
    <a:MessageID>urn:uuid:64f95d31-e078-4f2e-8bb2-d8e6e183alf0</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <o:Security s:mustUnderstand="1">
      <u:Timestamp u:Id="_0">
        <u:Created>2009-09-24T17:34:08Z</u:Created>
        <u:Expires>2009-09-24T17:39:08Z</u:Expires>
      </u:Timestamp>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
          <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
          <Reference URI="#_1">
            <Transforms>
              <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <DigestValue>Y6HYkPrH5NqSrdcLg8AYXDphZ74=</DigestValue>
          </Reference>
          <Reference URI="#_0">

```

```

    <Transforms>
      <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
    <DigestValue>1Taikh1jTPazJ2KnVddUmByNd/s</DigestValue>
  </Reference>
</SignedInfo>
<SignatureValue>dbpePnJ3w7i6Ro09jhxd60HKt3ssZPuSWVk ... ==</SignatureValue>
<KeyInfo>
  <o:SecurityTokenReference>
    <o:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-x509-token-profile-
1.0#X509SubjectKeyIdentifier">sUwVAnqj8qmOw5IJ7L0Z7s8fEh4=</o:KeyIdentifier>
  </o:SecurityTokenReference>
</KeyInfo>
</Signature>
</o:Security>
</s:Header>
<s:Body>
  <t:RequestSecurityToken Id="uuid-e067aa03-623a-4120-b8d9-64b60e8f1104">
    <t:RequestType>http://schemas.xmlsoap.org/ws/2005/02/trust/Issue</t:RequestType>
    <t:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV1.1</t:TokenType>
    <t:KeyType>http://schemas.xmlsoap.org/ws/2005/02/trust/SymmetricKey</t:KeyType>
    <t:KeySize>256</t:KeySize>
    <t:CanonicalizationAlgorithm>http://www.w3.org/2001/10/xml-exc-
c14n#</t:CanonicalizationAlgorithm>
    <t:EncryptionAlgorithm>http://www.w3.org/2001/04/xmlenc#aes256-
cbc</t:EncryptionAlgorithm>
    <t:EncryptWith>http://www.w3.org/2001/04/xmlenc#aes256-cbc</t:EncryptWith>
    <t:SignWith>http://www.w3.org/2000/09/xmldsig#hmac-sha1</t:SignWith>

    <t:ComputedKeyAlgorithm>http://schemas.xmlsoap.org/ws/2005/02/trust/CK/PSHA1</t:ComputedKeyAl
gorithm>
    <wsp:AppliesTo>
      <a:EndpointReference>
        <a:Address>http://fabrikam.com</a:Address>
      </a:EndpointReference>
    </wsp:AppliesTo>
    <t:OnBehalfOf>
      <saml:Assertion MajorVersion="1" MinorVersion="1" AssertionID="saml-6c5a4142-8257-
4efa-8b45-491feee53159" Issuer="contoso.com" IssueInstant="2009-09-24T17:34:09.095Z"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
        <saml:Conditions NotBefore="2009-09-24T17:34:09.079Z" NotOnOrAfter="2009-09-
24T17:39:09.079Z">
          <saml:AudienceRestrictionCondition>
            <saml:Audience>uri:WindowsLiveID</saml:Audience>
          </saml:AudienceRestrictionCondition>
        </saml:Conditions>
        <saml:AttributeStatement>
          <saml:Subject>
            <saml:NameIdentifier
Format="http://schemas.microsoft.com/LiveID/Federation/2008/05/ImmutableID">A0/HqOjr7EOU8HUUv
2Tgfg=@contoso.com</saml:NameIdentifier>
            <saml:SubjectConfirmation>
              <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-
vouches</saml:ConfirmationMethod>
            </saml:SubjectConfirmation>
          </saml:Subject>
        </saml:AttributeStatement>
      </saml:Assertion>
    </t:OnBehalfOf>
  </t:RequestSecurityToken>

```

```

        <saml:Attribute AttributeName="EmailAddress"
AttributeNamespace="http://schemas.xmlsoap.org/ws/2005/05/identity/claims">
        <saml:AttributeValue>joe@contoso.com</saml:AttributeValue>
        </saml:Attribute>
    </saml:AttributeStatement>
    <saml:AuthenticationStatement
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password" AuthenticationInstant="2009-
09-24T17:34:09.095Z">
        <saml:Subject>
            <saml:NameIdentifier
Format="http://schemas.microsoft.com/LiveID/Federation/2008/05/ImmutableID">A0/HqOjr7EOU8HUUV
2Tgfg==@contoso.com</saml:NameIdentifier>
            <saml:SubjectConfirmation>
                <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:sender-
vouches</saml:ConfirmationMethod>
            </saml:SubjectConfirmation>
        </saml:Subject>
    </saml:AuthenticationStatement>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
            <Reference URI="#saml-6c5a4142-8257-4efa-8b45-491fee53159">
                <Transforms>
                    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature" />
                    <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
                </Transforms>
                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
                <DigestValue>2fQF5XM8cqkXR/D0d/TigD3c6YM=</DigestValue>
            </Reference>
        </SignedInfo>
        <SignatureValue>b+MQeAJwIKGjoWgkE1+ookJ626nZ5 ... ==</SignatureValue>
        <KeyInfo>
            <o:SecurityTokenReference xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
                <o:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-x509-token-profile-
1.0#X509SubjectKeyIdentifier">sUwVANqj8qmOw5IJ7L0Z7s8fEh4=</o:KeyIdentifier>
            </o:SecurityTokenReference>
        </KeyInfo>
    </Signature>
</saml:Assertion>
</t:OnBehalfOf>
<auth:AdditionalContext>
    <auth:ContextItem
Scope="http://schemas.xmlsoap.org/ws/2006/12/authorization/ctx/requestor"
Name="http://schemas.microsoft.com/wlid/requestor">
        <auth:Value>contoso.com</auth:Value>
    </auth:ContextItem>
</auth:AdditionalContext>
<t:Claims Dialect="http://schemas.xmlsoap.org/ws/2006/12/authorization/authclaims">
    <auth:ClaimType
Uri="http://schemas.xmlsoap.org/ws/2006/12/authorization/claims/action">
        <auth:Value>MSExchange.SharingCalendarFreeBusy</auth:Value>
    </auth:ClaimType>
</t:Claims>
    <wsp:PolicyReference URI="EX_MBI_FED_SSL"></wsp:PolicyReference>
</t:RequestSecurityToken>
</s:Body>

```

</s:Envelope>

The following describes the required attributes and elements that are used in the example:

/s:Envelope/s:Header/a:To: The URI in this element is taken from the </FederationMetadata/Federation/TargetServiceEndpoint> element of the federation metadata document provided by the STS.

/s:Envelope/s:Header/o:Security/u:Timestamp/u:Created: The UTC time at which the request is made.

/s:Envelope/s:Header/o:Security/u:Timestamp/u:Expires: The UTC time at which the offer for the authentication token expires. This is the create time plus a duration. [<6>](#)

/s:Envelope/s:Header/o:Security/Signature : The standard signature of the <To> and <Timestamp> headers, as specified in [\[XMLDSIG\]](#).

/s:Envelope/s:Header/o:Security/Signature/Reference/DigestValue: The digest value that is returned by the specified digest method of the previous <To> and <Timestamp> headers, as specified in [\[XMLDSIG\]](#).

/s:Envelope/s:Header/o:Security/Signature/SignatureValue: The signature of the <To> and <Timestamp> headers, as specified in [\[XMLDSIG\]](#).

/s:Envelope/s:Header/o:Security/Signature/KeyInfo/o:SecurityTokenReference/o:KeyIdentifier: The <SubjectKeyIdentifier> value of the X509 certificate that is associated with the organization and sent to the STS by using the **CreateAppID** operation (section [3.1.4.2](#)) or **UpdateAppIdCertificate** operation (section [3.1.4.7](#)).

/s:Envelope/s:Body/t:RequestSecurityToken/wsp:AppliesTo/a:EndpointReference/a:Address: The URI of the organization to which the token will be sent.

/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:Assertion: Attributes of the <saml:Assertion> element, as shown in the following table.

| Attribute | Value |
|--------------|---|
| AssertionId | A unique identifier that identifies this specific token request. |
| Issuer | The URI of the organization that is requesting the token. This URI is the same as the value that is sent to the STS with the AddUri operation (section 3.1.4.1) <7> . |
| IssueInstant | The UTC date and time that the request is made. |

/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:Conditions: Attributes of the <saml:Conditions> element, as shown in the following table.

| Attribute | Value |
|--------------|---|
| NotBefore | The UTC date and time that the request is made. |
| NotOnOrAfter | The UTC date and time that the offer expires. |

/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:Conditions/saml:AudienceRestrictionCondition/saml:Audience: MUST be set to the URI of the STS. [<8>](#)

/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:AttributeStatement/saml:Subject/saml:NameIdentifier: The **Format** attribute of the <saml:NameIdentifier> element MUST be set to an identifier of the user for whom the token is requested. <9>

/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:AttributeStatement/saml:Attribute/ : An attribute MUST be set to the e-mail address of the user for whom the token is requested. The **AttributeName** attribute MUST be "EmailAddress".

/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:AttributeStatement/saml:Attribute/saml:AttributeValue: The e-mail address of the user for whom the token is requested. The domain part of the e-mail address MUST be one of the URI values previously registered with the **AddUri** operation (3.1.4.1).

/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:AuthenticationStatement/saml:Subject/saml:NameIdentifier: The **Format** attribute of the <saml:NameIdentifier> element MUST be set to an identifier of the user for whom the token is requested. The identifier MUST be the same as the **/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:AttributeStatement/saml:Subject/saml:NameIdentifier** element value. <10>

/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:AuthenticationStatement/saml:Signature: The <Signature> element is set to the standard XML signature of the <OnBehalfOf> element, as specified in [XMLDSIG]. Expected values for elements of the <Signature> element are as follows:

/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:AuthenticationStatement/saml:Signature/KeyInfo/o:KeyIdentifier: MUST be the <SubjectKeyIdentifier> element of the X509 certificate that is used when calling the **CreateAppId** operation (section 3.1.4.2).

/s:Envelope/s:Body/t:RequestSecurityToken/auth:AdditionalContext/auth:ContextItem: A <ContextItem> element with the **Scope** attribute set to "http://schemas.xmlsoap.org/ws/2006/12/authorization/ctx/requestor" and the name element set to "http://schemas.microsoft.com/wlid/requestor" MUST be present.

/s:Envelope/s:Body/t:RequestSecurityToken/auth:AdditionalContext/auth:ContextItem/auth:Value: MUST be set to the same URI as the value used for the **Issuer** attribute of the **/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:Assertion** element.

/s:Envelope/s:Body/t:RequestSecurityToken/t:Claims: The request MUST contain a <t:Claims> element with the **Dialect** attribute value set to "http://schemas.xmlsoap.org/ws/2006/12/authorization/authclaims" and containing at least one <auth:ClaimType> element.

/s:Envelope/s:Body/t:RequestSecurityToken/t:Claims/auth:ClaimType: The request MUST contain an <auth:ClaimType> element with the **Uri** attribute value set to "http://schemas.xmlsoap.org/ws/2006/12/authorization/claims/action" and containing at least one <<auth:Value>> element.

/s:Envelope/s:Body/t:RequestSecurityToken/t:Claims/auth:ClaimType/auth:Value: MUST be set to the name of the token offered. Can be any one of the following names:

- MExchange.SharingInviteMessage
- MExchange.SharingCalendarFreeBusy

- MExchange.SharingRead
- MExchange.DeliveryExternalSubmit
- MExchange.DeliveryInternalSubmit
- MExchange.MailboxMove
- MExchange.Autodiscover
- MSRMS.CertificationWS
- MSRMS.LicensingWS

/s:Envelope/s:Body/t:RequestSecurityToken/wsp:PolicyReference: The request MUST contain one <wsp:Policy> element with the URI attribute value set to the token policy to use. [<11>](#)

Token Response

The following is an example of the token response that is sent to an STS.

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:Action xmlns:S="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Action"
S:mustUnderstand="1">http://schemas.xmlsoap.org/ws/2005/02/trust/RSTR/Issue</wsa:Action>
    <wsa:To xmlns:S="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="To"
S:mustUnderstand="1">http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
    <wsse:Security S:mustUnderstand="1">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="TS">
        <wsu:Created>2009-09-24T17:34:01Z</wsu:Created>
        <wsu:Expires>2009-09-24T17:39:01Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </S:Header>
  <S:Body>
    <wst:RequestSecurityTokenResponse xmlns:S="http://www.w3.org/2003/05/soap-envelope"
xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:psf="http://schemas.microsoft.com/Passport/SoapServices/SOAPFault">
      <wst:TokenType>urn:oasis:names:tc:SAML:1.0</wst:TokenType>
      <wsp:AppliesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
        <wsa:EndpointReference>
          <wsa:Address>http://fabrikam.com</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <wst:Lifetime>
        <wsu:Created>2009-09-24T17:34:01Z</wsu:Created>
```

```

    <wsu:Expires>2009-10-09T17:34:01Z</wsu:Expires>
  </wst:Lifetime>
  <wst:RequestedSecurityToken>
    <EncryptedData xmlns="http://www.w3.org/2001/04/xmlenc#" Id="Assertion0"
Type="http://www.w3.org/2001/04/xmlenc#Element">
      <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-
cbc"></EncryptionMethod>
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <EncryptedKey>
          <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-
mgflp"></EncryptionMethod>
          <ds:KeyInfo Id="keyinfo">
            <wsse:SecurityTokenReference>
              <wsse:KeyIdentifier EncodingType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
1.0#X509SubjectKeyIdentifier">sUwVAnqj8qmOw5IJ7L0Z7s8fEh4=</wsse:KeyIdentifier>
            </wsse:SecurityTokenReference>
          </ds:KeyInfo>
        <CipherData>
          <CipherValue>mfYn2OYAGs6YaXw5P8L79mmHvHbd3+Of1QWprAmRww/Finek03IEa/r7LlxxGfb7FAA+ScthkQA...
==</CipherValue>
        </CipherData>
      </EncryptedKey>
    </ds:KeyInfo>
  </CipherData>
</CipherValue>
</wst:RequestedSecurityToken>
<wst:RequestedAttachedReference>
  <wsse:SecurityTokenReference>
    <wsse:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.0#SAMLAssertionID">uuid-c3a658d0-d832-43dc-bf57-2bfba93c13e5</wsse:KeyIdentifier>
  </wsse:SecurityTokenReference>
</wst:RequestedAttachedReference>
<wst:RequestedUnattachedReference>
  <wsse:SecurityTokenReference>
    <wsse:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.0#SAMLAssertionID">uuid-c3a658d0-d832-43dc-bf57-2bfba93c13e5</wsse:KeyIdentifier>
  </wsse:SecurityTokenReference>
</wst:RequestedUnattachedReference>
<wst:RequestedProofToken>
  <wst:BinarySecret>TfKqVImHiUlePfaBrAE6P6Jevxwl/XF8</wst:BinarySecret>
</wst:RequestedProofToken>
</wst:RequestSecurityTokenResponse>
</S:Body>
</S:Envelope>

```

The following describes the required attributes and elements that are used in the example:

/s:body/wst:RequestSecurityTokenResponse: The response from the server MUST contain at least one <wst:RequestSecurityTokenResponse> element, as specified in [\[WSTRUST\]](#), with child elements as described as follows:

/s:body/wst:RequestSecurityTokenResponse/wsp:AppliesTo: The response MUST contain the <wsp:AppliesTo> element with at least one child <wsa:EndpointReference> element.

/s:body/wst:RequestSecurityTokenResponse/wsp:AppliesTo/wsa:EndpointReference: The response MUST contain the <wsa:EndpointReference> element with at least one child <wsa:Address> element.

/s:body/wst:RequestSecurityTokenResponse/wsp:AppliesTo/wsa:EndpointReference/wsa:Address: The <wsa:Address> element MUST contain the same value as the **/s:Envelope/s:Body/t:RequestSecurityToken/wsp:AppliesTo/a:EndpointReference/a:Address** element specified in the token request.

/s:body/wst:RequestSecurityTokenResponse/wst:RequestedSecurityToken: The response MUST contain at most one <wst:RequestedSecurityToken> element that MUST contain one and only one <EncryptedData> child element that contains the encrypted token that will be sent to another service for authentication. For more information about the contents of the token, see section [4.2.2](#).

/s:body/wst:RequestSecurityTokenResponse/wst:RequestedAttachedReference: The response MUST contain at least one wst:RequestedAttachedReference element that contains at least one child <wsse:SecurityTokenReference> element.

/s:body/wst:RequestSecurityTokenResponse/wst:RequestedAttachedReference/wsse:SecurityTokenReference: The response MUST contain at least one <wsse:SecurityTokenReference> element that contains at least one child <wsse:KeyIdentifier> element.

/s:body/wst:RequestSecurityTokenResponse/wst:RequestedAttachedReference/wsse:SecurityTokenReference/wsse:KeyIdentifier: The response MUST contain at least one <wsse:KeyIdentifier> element that contains the identifier of the SAML assertion encrypted within the <RequestedSecurityToken> element.

/s:body/wst:RequestSecurityTokenResponse/wst:RequestedProofToken: The response MUST contain at least one wst:RequestedProofToken element that contains at least one child <wst:BinarySecret> element.

/s:body/wst:RequestSecurityTokenResponse/wst:RequestedAttachedReference/wst:RequestedProofToken/wst:BinarySecret: The response MUST contain a <wst:BinarySecret> element with the value set to the symmetric key that is encrypted in the <RequestedSecurityToken> element.

4.2.2 Encrypted and Unencrypted Tokens

This section shows the encrypted and unencrypted tokens that are received from the Secure Token Service.

Encrypted Token

The following is an example of the encrypted token that is received from an STS.

```
<EncryptedData xmlns="http://www.w3.org/2001/04/xmlenc#" Id="Assertion0"
Type="http://www.w3.org/2001/04/xmlenc#Element">
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-
cbc"></EncryptionMethod>
  <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <EncryptedKey>
```

```

    <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-
mgflp"></EncryptionMethod>
    <ds:KeyInfo Id="keyinfo">
      <wsse:SecurityTokenReference>
        <wsse:KeyIdentifier EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
1.0#X509SubjectKeyIdentifier">sUwVANqj8qmOw5IJ7L0Z7s8fEh4=</wsse:KeyIdentifier>
      </wsse:SecurityTokenReference>
    </ds:KeyInfo>
    <CipherData>
      <CipherValue>mfYn2OYAGs6YaXw5P8L79mmHvHbd3+Of1QWprAmRww/Finek03IEa/r7LlxxGfb7FAA+ScthkQA...
==</CipherValue>
    </CipherData>
  </EncryptedKey>
</ds:KeyInfo>
<CipherData>
  <CipherValue>B5B4B/PrdcBj9s8CQxBs6pNNLFLA9VeA4Y5ZIM6VBkDYwX6zmnCmBkOghx9pPrSGxmp2KChWU5QAKHsJ
...==</CipherValue>
  </CipherData>
</EncryptedData>

```

Unencrypted Token

The following is an example of the unencrypted token that is received from an STS.

```

<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" AssertionID="uuid-
c3a658d0-d832-43dc-bf57-2bfba93c13e5" IssueInstant="2009-09-24T17:34:01Z"
Issuer="uri:WindowsLiveID" MajorVersion="1" MinorVersion="1">
  <saml:Conditions NotBefore="2009-09-24T17:34:01Z" NotOnOrAfter="2009-10-09T17:34:01Z">
    <saml:AudienceRestrictionCondition>
      <saml:Audience>http://fabrikam.com</samlAudience >
    </saml:AudienceRestrictionCondition>
  </saml:Conditions>
  <saml:AuthenticationStatement AuthenticationInstant="2009-09-24T17:34:01Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
    <saml:Subject>
      <saml:NameIdentifier
Format="http://schemas.xmlsoap.org/claims/UPN">a744b0351351444d3087ca806986b9a0@Live.com</sam
l:NameIdentifier>
      <saml:SubjectConfirmation>
        <saml:ConfirmationMethod>urn:oasis:names:tc:saml:1.0:cm:holder-of-
key</saml:ConfirmationMethod>
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
          <e:EncryptedKey xmlns:e="http://www.w3.org/2001/04/xmlenc#">
            <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-
mgflp"></e:EncryptionMethod>
            <ds:KeyInfo Id="keyinfo">
              <wsse:SecurityTokenReference xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
                <wsse:KeyIdentifier EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
1.0#X509SubjectKeyIdentifier">sUwVANqj8qmOw5IJ7L0Z7s8fEh4=</wsse:KeyIdentifier>
              </wsse:SecurityTokenReference>
            </ds:KeyInfo>
          </e:EncryptedKey>
        </ds:KeyInfo>
      </saml:SubjectConfirmation>
    </saml:Subject>
  </saml:AuthenticationStatement>
</saml:Assertion>

```

```

</ds:KeyInfo>
<e:CipherData>
  <e:CipherValue>lRRb1PaUiQrsdA0me/Q4Gt6RVHkDm5ehPNZaDoiQ ... ==</e:CipherValue>
</e:CipherData>
</e:EncryptedKey>
  </ds:KeyInfo>
</saml:SubjectConfirmation>
</saml:Subject>
</saml:AuthenticationStatement>
<saml:AttributeStatement>
  <saml:Subject>
    <saml:NameIdentifier
Format="http://schemas.xmlsoap.org/claims/UPN">a744b0351351444d3087ca806986b9a0@Live.com</sam
l:NameIdentifier>
    </saml:Subject>
    <saml:Attribute AttributeName="RequestorDomain"
AttributeName="http://schemas.microsoft.com/ws/2006/04/identity/claims">
      <saml:AttributeValue>contoso.com</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute AttributeName="EmailAddress"
AttributeName="http://schemas.xmlsoap.org/claims">
      <saml:AttributeValue>joe@contoso.com</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute AttributeName="action"
AttributeName="http://schemas.xmlsoap.org/ws/2006/12/authorization/claims">
      <saml:AttributeValue>MSExchange.SharingCalendarFreeBusy</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute AttributeName="ThirdPartyRequested"
AttributeName="http://schemas.microsoft.com/ws/2006/04/identity/claims">
      <saml:AttributeValue></saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute AttributeName="AuthenticatingAuthority"
AttributeName="http://schemas.microsoft.com/ws/2008/06/identity">
      <saml:AttributeValue>http://contoso.com</saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></CanonicalizationMethod>
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"></SignatureMethod>
      <Reference URI="#uuid-c3a658d0-d832-43dc-bf57-2bfba93c13e5">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"></Transform>
          <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"></Transform>
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
        <DigestValue>DP2Bg6+h59Uw4zc8DjRNJ4UQAlw=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>
      baY0k5dLPuPHKcWtgMATaXKEJL4vX8GeWvaQgCeZchNUbXij1BmPH/Lqu/1HtFavGpLDJ+ukbGeV
      vKWveIGCnre8SCYBUBHlwi0FSw+p+pmFGLRytRG4mkAzEI9dskGnW0RlhFfSVDzvnSBGwrNzSH5o
      Y9hKDVT5emRGeYpDQYc=
    </SignatureValue>
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="keyinfo">
      <wsse:SecurityTokenReference xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">

```

```

    <wsse:KeyIdentifier EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
1.0#X509SubjectKeyIdentifier">VbJyIcGL0AjB4/Wm4DgUZux6uUk=</wsse:KeyIdentifier>
    </wsse:SecurityTokenReference>
  </ds:KeyInfo>
</Signature>
</saml:Assertion>

```

The following describes the required attributes and elements that are used in the example:

/saml:Assertion: The <AssertionID> attribute value MUST match the **/s:body/wst:RequestSecurityTokenResponse/wst:RequestedAttachedReference/wse:SecurityTokenReference/wsse:KeyIdentifier** element in the response from the STS.

/saml:Assertion/saml:Conditions/saml:AudienceRestrictionCondition/saml:Audience: The <saml:Audience> element MUST contain the same value as the **/s:Envelope/s:Body/t:RequestSecurityToken/wsp:AppliesTo/a:EndpointReference/a:Address** element in the request.

/saml:Assertion/saml:AuthenticationStatement/saml:Subject/saml:NameIdentifier: The <saml:NameIdentifier> element MUST be present and MUST be in UPN syntax, but can be any value that the STS wants; however, it must always be the same for each **/s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:AuthenticationStatement/saml:Subject/saml:NameIdentifier** element in the request.

/saml:Assertion/saml:AuthenticationStatement/saml:Subject/saml:SubjectConfirmation: The <saml:SubjectConfirmation> element MUST be present and MUST be in the format specified in [\[SAML\]](#).

/saml:Assertion/saml:AttributeStatement/saml:Subject/saml:NameIdentifier: The value of the <saml:NameIdentifier> element MUST be the same as the **/saml:Assertion/saml:AuthenticationStatement/saml:Subject/saml:NameIdentifier** element.

/saml:Assertion/saml:AttributeStatement/saml:Attribute: The <saml:AttributeStatement> element MUST contain the following <Attribute> elements:

| Attribute name | <AttributeValue> element |
|---------------------|---|
| RequestorDomain | MUST be the same as the /s:Envelope/s:Body/t:RequestSecurityToken/auth:AdditionalContext/auth:ContextItem/auth:Value element in the token request. |
| EmailAddresses | MUST be the same as the /s:Envelope/s:Body/t:RequestSecurityToken/t:OnBehalfOf/saml:Assertion/saml:AttributeStatement/saml:Attribute@[EmailAddresses]\AttributeValue element in the token request. |
| action | MUST be the same as the /s:Envelope/s:Body/t:RequestSecurityToken/t:Claims\auth:ClaimType@[.../Action]\auth:Value element in the token request. |
| ThirdPartyRequested | MUST NOT contain a value. |

| Attribute name | <AttributeValue> element |
|-------------------------|---|
| AuthenticatingAuthority | MUST contain a domain name previously registered with the AddUri operation (section 3.1.4.1). |

/saml:Assertion/Signature: The <Signature> element MUST be a standard signature, as specified in [XMLDSIG](#), and MUST sign the entire <Assertion> element.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

The following is the WSDL file that defines the Manage Delegation Web service.

```
<?xml version="1.0" encoding="utf8" ?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://domains.live.com/Service/ManageDelegation/V1.0"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://domains.live.com/Service/ManageDelegation/V1.0"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://domains.live.com/Service/ManageDelegation/V1.0">
      <s:element name="CreateAppId">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="certificate" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="properties"
type="tns:ArrayOfProperty" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfProperty">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="Property" type="tns:Property" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="Property">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="Value" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="CreateAppIdResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="CreateAppIdResult"
type="tns:AppIdInfo" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="AppIdInfo">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="AppId" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="AdminKey" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="UpdateAppIdCertificate">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="appId" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>

```

```

        <s:element minOccurs="0" maxOccurs="1" name="appIdAdminKey" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="newCertificate" type="s:string" />
    </s:sequence>
</s:complexType>
</s:element>
<s:element name="UpdateAppIdCertificateResponse">
    <s:complexType />
</s:element>
<s:element name="UpdateAppIdProperties">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="appId" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="properties"
type="tns:ArrayOfProperty" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="UpdateAppIdPropertiesResponse">
    <s:complexType />
</s:element>
<s:element name="AddUri">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ownerAppId" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="uri" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="AddUriResponse">
    <s:complexType />
</s:element>
<s:element name="RemoveUri">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ownerAppId" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="uri" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="RemoveUriResponse">
    <s:complexType />
</s:element>
<s:element name="ReserveDomain">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ownerAppId" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="domainName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="programId" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="ReserveDomainResponse">
    <s:complexType />
</s:element>
<s:element name="ReleaseDomain">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ownerAppId" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="domainName" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>

```

```

        </s:sequence>
    </s:complexType>
</s:element>
    <s:element name="ReleaseDomainResponse">
        <s:complexType />
    </s:element>
    <s:element name="GetDomainInfo">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1" name="ownerAppId" type="s:string" />
                <s:element minOccurs="0" maxOccurs="1" name="domainName" type="s:string" />
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="GetDomainInfoResponse">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1" name="GetDomainInfoResult"
type="tns:DomainInfo" />
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:complexType name="DomainInfo">
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="DomainName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="AppId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="DomainState" type="tns:DomainState" />
        </s:sequence>
    </s:complexType>
    <s:simpleType name="DomainState">
        <s:restriction base="s:string">
            <s:enumeration value="PendingActivation" />
            <s:enumeration value="Active" />
            <s:enumeration value="PendingRelease" />
        </s:restriction>
    </s:simpleType>
</s:schema>
</wsdl:types>
    <wsdl:message name="CreateAppIdSoapIn">
        <wsdl:part name="parameters" element="tns:CreateAppId" />
    </wsdl:message>
    <wsdl:message name="CreateAppIdSoapOut">
        <wsdl:part name="parameters" element="tns:CreateAppIdResponse" />
    </wsdl:message>
    <wsdl:message name="UpdateAppIdCertificateSoapIn">
        <wsdl:part name="parameters" element="tns:UpdateAppIdCertificate" />
    </wsdl:message>
    <wsdl:message name="UpdateAppIdCertificateSoapOut">
        <wsdl:part name="parameters" element="tns:UpdateAppIdCertificateResponse" />
    </wsdl:message>
    <wsdl:message name="UpdateAppIdPropertiesSoapIn">
        <wsdl:part name="parameters" element="tns:UpdateAppIdProperties" />
    </wsdl:message>
    <wsdl:message name="UpdateAppIdPropertiesSoapOut">
        <wsdl:part name="parameters" element="tns:UpdateAppIdPropertiesResponse" />
    </wsdl:message>
    <wsdl:message name="AddUriSoapIn">
        <wsdl:part name="parameters" element="tns:AddUri" />
    </wsdl:message>

```

```

    <wsdl:message name="AddUriSoapOut">
      <wsdl:part name="parameters" element="tns:AddUriResponse" />
    </wsdl:message>
    <wsdl:message name="RemoveUriSoapIn">
      <wsdl:part name="parameters" element="tns:RemoveUri" />
    </wsdl:message>
    <wsdl:message name="RemoveUriSoapOut">
      <wsdl:part name="parameters" element="tns:RemoveUriResponse" />
    </wsdl:message>
    <wsdl:message name="ReserveDomainSoapIn">
      <wsdl:part name="parameters" element="tns:ReserveDomain" />
    </wsdl:message>
    <wsdl:message name="ReserveDomainSoapOut">
      <wsdl:part name="parameters" element="tns:ReserveDomainResponse" />
    </wsdl:message>
    <wsdl:message name="ReleaseDomainSoapIn">
      <wsdl:part name="parameters" element="tns:ReleaseDomain" />
    </wsdl:message>
    <wsdl:message name="ReleaseDomainSoapOut">
      <wsdl:part name="parameters" element="tns:ReleaseDomainResponse" />
    </wsdl:message>
    <wsdl:message name="GetDomainInfoSoapIn">
      <wsdl:part name="parameters" element="tns:GetDomainInfo" />
    </wsdl:message>
    <wsdl:message name="GetDomainInfoSoapOut">
      <wsdl:part name="parameters" element="tns:GetDomainInfoResponse" />
    </wsdl:message>
    <wsdl:portType name="ManageDelegationSoap">
      <wsdl:operation name="CreateAppId">
        <wsdl:input message="tns:CreateAppIdSoapIn" />
        <wsdl:output message="tns:CreateAppIdSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="UpdateAppIdCertificate">
        <wsdl:input message="tns:UpdateAppIdCertificateSoapIn" />
        <wsdl:output message="tns:UpdateAppIdCertificateSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="UpdateAppIdProperties">
        <wsdl:input message="tns:UpdateAppIdPropertiesSoapIn" />
        <wsdl:output message="tns:UpdateAppIdPropertiesSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="AddUri">
        <wsdl:input message="tns:AddUriSoapIn" />
        <wsdl:output message="tns:AddUriSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="RemoveUri">
        <wsdl:input message="tns:RemoveUriSoapIn" />
        <wsdl:output message="tns:RemoveUriSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="ReserveDomain">
        <wsdl:input message="tns:ReserveDomainSoapIn" />
        <wsdl:output message="tns:ReserveDomainSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="ReleaseDomain">
        <wsdl:input message="tns:ReleaseDomainSoapIn" />
        <wsdl:output message="tns:ReleaseDomainSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="GetDomainInfo">
        <wsdl:input message="tns:GetDomainInfoSoapIn" />
        <wsdl:output message="tns:GetDomainInfoSoapOut" />
      </wsdl:operation>

```

```

        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="ManageDelegationSoap" type="tns:ManageDelegationSoap">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
        <wsdl:operation name="CreateAppId">
            <soap:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/CreateAppId"
style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="UpdateAppIdCertificate">
            <soap:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/UpdateAppIdCertificate"
style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="UpdateAppIdProperties">
            <soap:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/UpdateAppIdProperties"
style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="AddUri">
            <soap:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/AddUri" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="RemoveUri">
            <soap:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/RemoveUri" style="document"
/>
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="ReserveDomain">

```

```

    <soap:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/ReserveDomain"
style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ReleaseDomain">
    <soap:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/ReleaseDomain"
style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetDomainInfo">
    <soap:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/GetDomainInfo"
style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ManageDelegationSoap12" type="tns:ManageDelegationSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="CreateAppId">
        <soap12:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/CreateAppId"
style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateAppIdCertificate">
        <soap12:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/UpdateAppIdCertificate"
style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateAppIdProperties">

```

```

    <soap12:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/UpdateAppIdProperties"
style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="AddUri">
    <soap12:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/AddUri" style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RemoveUri">
    <soap12:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/RemoveUri" style="document"
/>
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ReserveDomain">
    <soap12:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/ReserveDomain"
style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ReleaseDomain">
    <soap12:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/ReleaseDomain"
style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetDomainInfo">
    <soap12:operation
soapAction="http://domains.live.com/Service/ManageDelegation/V1.0/GetDomainInfo"
style="document" />
    <wsdl:input>
        <soap12:body use="literal" />
    </wsdl:input>

```



```
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="ManageDelegation">
      <wsdl:port name="ManageDelegationSoap" binding="tns:ManageDelegationSoap">
        <soap:address location="https://domains.live.com/service/managedelegation.asmx" />
      </wsdl:port>
      <wsdl:port name="ManageDelegationSoap12" binding="tns:ManageDelegationSoap12">
        <soap12:address location="https://domains.live.com/service/managedelegation.asmx" />
      </wsdl:port>
    </wsdl:service>
  </wsdl:definitions>
```

7 Appendix B: Product Behavior

The information in this specification is applicable to the following product versions. References to product versions include released service packs.

- Microsoft® Exchange Server 2010

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified.

Unless otherwise specified, any statement of optional behavior in this specification 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 product does not follow the prescription.

[<1> Section 1.5:](#) By default, Exchange server gets the Federation Metadata Document from the URL <http://nexus.passport.com/FederationMetadata/2006-12/FederationMetadata.xml>. This URL can be modified when establishing the federated domain by using Exchange server command-line tools.

[<2> Section 1.5:](#) Exchange Server 2010 stores the URL of the delegation management service in Active Directory when the server is loaded. The URL is stored in the **serviceBindingInformation** property of the object `CN=DomainPartnerManageDelegation,CN=ServiceEndpoints,CN=FirstOrganization,CN=MicrosoftExchange,CN=Services,CN=Configuration,DC=`

When the Exchange server calls the delegation management service, this object is read to obtain the URL of the service.

[<3> Section 3.1.4.6.1.1:](#) The Exchange server sets this element to the string "ExchangeConnector."

[<4> Section 3.2:](#) The federation metadata document MUST contain the following XML elements and element values for Exchanger Server:

| Element | Description |
|-----------------------|---|
| <FederationMetadata> | MUST be present. MUST contain at least one <Federation> element. |
| <Federation> | MUST be present. MUST contain at least one of each of the following elements: <ul style="list-style-type: none">▪ <TokenSigningKeyInfo>▪ <IssuerNamesOffered>▪ <TargetServiceEndpoints>▪ <WebRequestorRedirectEndpoints> |
| <TokenSigningKeyInfo> | At least one instance MUST be present. MUST contain at least one <X509Certificate> element. The first instance MUST contain the Id attribute with the value "stscer". The second instance, if any, MUST contain the Id attribute with the value "stsbcer". |
| <X509Certificate> | MUST be present. |

| Element | Description |
|---------------------------------|---|
| <IssuerNamesOffered> | MUST be present. MUST contain the uri attribute with the value "uri:WindowsLiveId". |
| <TargetServiceEndpoints> | MUST be present. MUST contain at least one Address element which MUST contain a valid absolute path URI. |
| <WebRequestorRedirectEndpoints> | MUST be present. MUST contain at least one Address element which MUST contain a valid absolute path URI. |

<5> [Section 4.1.2:](#) The Exchange server sets this element to the string "ExchangeConnector."

<6> [Section 4.2.1:](#) The duration of the offer depends on the type of offer made. An Exchange server creates an offer with the duration set to the following values:

| Offer type | Default duration |
|-------------------------------------|------------------|
| MSEExchange.SharingInviteMessage | 15 days |
| MSEExchange.SharingCalendarFreeBusy | 5 minutes |
| MSEExchange.SharingRead | 60 minutes |
| MSEExchange.DeliveryExternalSubmit | 48 hours |
| MSEExchange.DeliveryInternalSubmit | 48 hours |
| MSEExchange.MailboxMove | 60 minutes |
| MSEExchange.Autodiscover | 5 minutes |

<7> [Section 4.2.1:](#) Exchange Server stores this value in the Active Directory property **msExchFedApplicationURI** of the **msExchFedTrust** object.

<8> [Section 4.2.1:](#) Exchange Server stores this value in the Active Directory property **msExchFedTokenIssuerURI** of the **msExchFedTrust** object. Exchange Server always uses the value "uri:WindowsLiveID".

<9> [Section 4.2.1:](#) Exchange Server obtains the value of the <saml:NameIdentifier> element from the user object in Active Directory of the user for whom the token is requested. If the Active Directory **user** object has the **msExchImmutable** property set, that value is used; otherwise, the Exchange server uses the base-64 encoded **objectGuid** property of the user object concatenated with the **msExchFedAccountNamespace** property of the **msExchFedOrgId** object.

<10> [Section 4.2.1:](#) Exchange Server obtains the value of the <saml:NameIdentifier> element from the user object in Active Directory of the user for whom the token is requested. If the Active Directory **user** object has the **msExchImmutable** property set, that value is used; otherwise the Exchange server uses the base-64 encoded **objectGuid** property of the user object concatenated with the **msExchFedAccountNamespace** property of the **msExchFedOrgId** object.

<11> [Section 4.2.1:](#) Exchange Server sets the URI to the attribute value found in the Active Directory property **msExchFedPolicyReferenceURI** of the **msExchFedTrust** object. The default value is "EX_MBI_FED_SSL".

8 Change Tracking

This section identifies changes made to [MS-OXWSLVID] protocol documentation between February 2010 and May 2010 releases. Changes are classed as major, minor, or editorial.

Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- A protocol is deprecated.
- The removal of a document from the documentation set.
- Changes made for template compliance.

Minor changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

Editorial changes apply to grammatical, formatting, and style issues.

No changes means that the document is identical to its last release.

Major and minor changes can be described further using the following revision types:

- New content added.
- Content update.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.

- Content removed for template compliance.
- Obsolete document removed.

Editorial changes always have the revision type "Editorially updated."

Some important terms used in revision type descriptions are defined as follows:

Protocol syntax refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

Protocol revision refers to changes made to a protocol that affect the bits that are sent over the wire.

Changes are listed in the following table. If you need further information, please contact protocol@microsoft.com.

| Section | Tracking number (if applicable) and description | Major change (Y or N) | Revision Type |
|---|--|-----------------------|--------------------|
| 1.1 Glossary | Added the terms "Uniform Resource Locator (URL)" and "Uniform Resource Identifier (URI)" to the list of terms that are defined in [MS-OXGLOS]. | N | Content update. |
| 1.2.1 Normative References | 54785 Added reference to [MS-OXWSMSHR]. | N | Content update. |
| 1.3 Overview | Updated the section title. | N | Content update. |
| 1.4 Relationship to Other Protocols | 54785 Added information to explain the relationship between this protocol and other protocols. | N | New content added. |
| 1.5 Prerequisites/Preconditions | Added a glossary term link for the term "URL". | N | Content update. |
| 3 Protocol Details | Moved content from Section 3.1 to this section. | N | Content update. |
| 3.1 ManageDelegationSoap Client Details | Added a glossary term link for the term "URI". | N | Content update. |
| 3.1.4.1 AddUri | Added a glossary term link for the terms "URL" and "URI". | N | Content update. |
| 3.1.4.1.1 AddUri Element | Added a glossary term link for the term "URI". | N | Content update. |
| 3.1.4.1.2.1 tns:AddUriSoapIn Message | Added a glossary term link for the term "URI". | N | Content update. |
| 8 Change Tracking | Removed section 3.1. | N | Content removed. |

9 Index

A

[AddUri](#) 13
[Applicability](#) 8
[ArrayOfPropertyType complex type](#) 10

C

[Change tracking](#) 60
[Complex types](#) 9

E

[Examples](#) 31

F

[Federation metadata client details](#) 29
[abstract data model](#) 29

I

[Introduction](#) 6

M

[ManageDelegationSoap client details](#) 12
[message processing events](#) 12

Messages
[syntax](#) 9
[transport](#) 9

N

[Namespaces](#) 9

O

[Overview](#) 8

P

[Preconditions](#) 8
[Prerequisites](#) 8
[Product behavior](#) 58
[Property complex type](#) 10

R

References
[normative](#) 6
[Relationship to other protocols](#) 8

T

[Tracking changes](#) 60