

[MS-ADFSWAP-Diff]:

Active Directory Federation Service (AD FS) Web Agent Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
10/24/2008	0.1	New	Version 0.1 release
12/5/2008	1.0	Major	Updated and revised the technical content.
1/16/2009	1.0.1	Editorial	Changed language and formatting in the technical content.
2/27/2009	2.0	Major	Updated and revised the technical content.
4/10/2009	2.0.1	Editorial	Changed language and formatting in the technical content.
5/22/2009	2.0.2	Editorial	Changed language and formatting in the technical content.
7/2/2009	2.0.3	Editorial	Changed language and formatting in the technical content.
8/14/2009	2.0.4	Editorial	Changed language and formatting in the technical content.
9/25/2009	2.0.5	Editorial	Changed language and formatting in the technical content.
11/6/2009	2.0.6	Editorial	Changed language and formatting in the technical content.
12/18/2009	2.0.7	Editorial	Changed language and formatting in the technical content.
1/29/2010	3.0	Major	Updated and revised the technical content.
3/12/2010	3.0.1	Editorial	Changed language and formatting in the technical content.
4/23/2010	3.0.2	Editorial	Changed language and formatting in the technical content.
6/4/2010	3.0.3	Editorial	Changed language and formatting in the technical content.
7/16/2010	4.0	Major	Updated and revised the technical content.
8/27/2010	4.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2010	5.0	Major	Updated and revised the technical content.
11/19/2010	5.0	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	5.0	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	5.0	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	5.0	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	5.0	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	5.1	Minor	Clarified the meaning of the technical content.
9/23/2011	5.1	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	6.0	Major	Updated and revised the technical content.

Date	Revision History	Revision Class	Comments
3/30/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	6.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	7.0	Major	Updated and revised the technical content.
11/14/2013	7.0	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	7.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	7.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	8.0	Major	Significantly changed the technical content.
7/14/2016	9.0	Major	Significantly changed the technical content.
6/1/2017	9.0	None	No changes to the meaning, language, or formatting of the technical content.
9/15/2017	10.0	Major	Significantly changed the technical content.
9/12/2018	11.0	Major	Significantly changed the technical content.
4/7/2021	12.0	Major	Significantly changed the technical content.
4/23/2024	13.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	7
1.2.1	(Updated Section) Normative References	7
1.2.2	Informative References	8
1.3	Overview	8
1.4	Relationship to Other Protocols	9
1.5	Prerequisites/Preconditions	9
1.6	Applicability Statement	10
1.7	Versioning and Capability Negotiation	10
1.7.1	Versioning	10
1.7.2	Capability Negotiation	10
1.8	Vendor-Extensible Fields	10
1.9	Standards Assignments	10
2	Messages	11
2.1	Transport	11
2.2	Common Message Syntax	11
2.2.1	Namespaces	11
2.2.2	Messages	11
2.2.3	Elements	11
2.2.4	Complex Types	11
2.2.5	Simple Types	12
2.2.6	Attributes	12
2.2.7	Groups	12
2.2.8	Attribute Groups	12
3	Protocol Details	13
3.1	Server Details	13
3.1.1	Abstract Data Model	13
3.1.1.1	GetFsTrustInformation	13
3.1.1.2	GetTrustedRealmUri	14
3.1.1.3	GetClaims	14
3.1.2	Timers	14
3.1.3	Initialization	15
3.1.4	Message Processing Events and Sequencing Rules	15
3.1.4.1	GetFsTrustInformation	15
3.1.4.1.1	Messages	15
3.1.4.1.1.1	GetFsTrustInformationSoapIn	15
3.1.4.1.1.1.1	SoftwareVersion Parameter	16
3.1.4.1.1.2	GetFsTrustInformationSoapIn Processing	16
3.1.4.1.1.3	GetFsTrustInformationSoapOut	16
3.1.4.1.1.3.1	SoftwareVersion Parameter	17
3.1.4.1.1.3.2	hostedRealmUri Parameter	17
3.1.4.1.1.3.3	IsUrl Parameter	17
3.1.4.1.1.3.4	fsDomainAccount Parameter	17
3.1.4.1.1.3.5	X509Thumbprint Parameter	18
3.1.4.1.1.3.6	SerializedStore Parameter	18
3.1.4.1.1.4	GetFsTrustInformationSoapOut Processing	18
3.1.4.1.1.4.1	Versioning Processing	19
3.1.4.1.1.4.2	Certificate Processing	19
3.1.4.1.1.4.3	Other Processing	19
3.1.4.2	GetTrustedRealmUri	19
3.1.4.2.1	Messages	19
3.1.4.2.1.1	GetTrustedRealmUriSoapIn	20
3.1.4.2.1.1.1	Email Parameter	20
3.1.4.2.1.2	GetTrustedRealmUriSoapIn Processing	20

3.1.4.2.1.3	GetTrustedRealmUriSoapOut	20
3.1.4.2.1.3.1	trustedRealmUri	20
3.1.4.2.1.4	GetTrustedRealmUriSoapOut Processing	20
3.1.4.3	GetClaims	21
3.1.4.3.1	Messages	21
3.1.4.3.1.1	GetClaimsSoapIn	21
3.1.4.3.1.2	GetClaimsSoapIn Processing	21
3.1.4.3.1.3	GetClaimsSoapOut	21
3.1.4.3.1.4	GetClaimsSoapOut Processing	22
3.1.5	Timer Events.....	22
3.1.6	Other Local Events.....	22
3.2	Client Details	23
3.2.1	Abstract Data Model.....	23
3.2.2	Timers	23
3.2.3	Initialization.....	23
3.2.3.1	GetFsTrustInformation Initialization.....	23
3.2.3.2	GetTrustedRealmUri Initialization	23
3.2.3.3	GetClaims Initialization	23
3.2.4	Message Processing Events and Sequencing Rules	23
3.2.4.1	GetFsTrustInformation.....	23
3.2.4.1.1	GetFsTrustInformationSoapIn Processing	24
3.2.4.1.2	GetFsTrustInformationSoapOut Processing	24
3.2.4.1.2.1	Versioning.....	24
3.2.4.1.2.2	Certificates	24
3.2.4.1.2.3	Other Data.....	25
3.2.4.2	GetTrustedRealmUri	25
3.2.4.2.1	GetTrustedRealmUriSoapIn Processing	25
3.2.4.2.2	GetTrustedRealmUriSoapOut Processing	25
3.2.4.3	GetClaims	25
3.2.4.3.1	GetClaimsSoapIn Processing	25
3.2.4.3.2	GetClaimsSoapOut Processing.....	25
3.2.5	Timer Events.....	26
3.2.6	Other Local Events.....	26
4	Protocol Examples	27
4.1	GetFsTrustInformation Request Message Example	27
4.2	GetFsTrustInformation Response Message Example.....	27
4.3	GetTrustedRealmUri Request Message Example	28
4.4	GetTrustedRealmUri Response Message Example	28
4.5	GetClaims Request Message Example	28
4.6	GetClaims Response Message Example	29
5	Security.....	30
5.1	Security Considerations for Implementers	30
5.2	Index of Security Parameters	30
6	Appendix A: Full WSDL	31
7	(Updated Section) Appendix B: Product Behavior.....	40
8	Change Tracking.....	41
9	Index.....	42

1 Introduction

The Active Directory Federation Service (AD FS) Web Agent Protocol is used by a Web service (WS) resource to obtain configuration data about a security token service (STS) in order to validate security tokens from that STS using the protocol defined in [MS-MWBF].

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

1.1 Glossary

This document uses the following terms:

claim: A declaration made by an entity (for example, name, identity, key, group, privilege, and capability). For more information, see [WSFederation1.2].

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the GUID. See also universally unique identifier (UUID).

security token: A collection of one or more claims. Specifically in the case of mobile devices, a security token represents a previously authenticated user as defined in the Mobile Device Enrollment Protocol [MS-MDE].

security token service (STS): A web service that issues security tokens. That is, it makes assertions based on evidence that it trusts; these assertions are for consumption by whoever trusts it.

web service (WS) resource: A destination HTTP 1.1 web application or an HTTP 1.1 resource serviced by the application. In the context of this protocol, it refers to the application or manager of the resource that receives identity information and assertions issued by an IP/STS using this protocol. The WS resource is a relying party in the context of this protocol. For more information, see [WSFederation1.2].

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

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

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

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

XML Schema (XSD): A language that defines the elements, attributes, namespaces, and data types for XML documents as defined by [XMLSCHEMA1/2] and [XMLSCHEMA2/2] standards. An XML schema uses XML syntax for its language.

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

1.2 References

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

1.2.1 (Updated Section) Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

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

[MS-MWBF] Microsoft Corporation, "Microsoft Web Browser Federated Sign-On Protocol".

[RFC1738] Berners-Lee, T., Masinter, L., and McCahill, M., Eds., "Uniform Resource Locators (URL)", RFC 1738, December 1994, <http://www.rfc-editor.org/rfcinfo/rfc1738.txt>

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

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

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

[RFC2822] Resnick, P., Ed., "Internet Message Format", RFC 2822, April 2001, <http://www.rfc-editor.org/rfcinfo/rfc2822.txt>

[RFC3852] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004, <http://www.rfc-editor.org/rfcinfo/rfc3852.txt>

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006, <http://www.rfc-editor.org/rfcinfo/rfc4648.txt>

[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/>

[SOAP1.2-1/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", W3C Recommendation, April 2007, <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>

[SOAP1.2-2/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 2: Adjuncts (Second Edition)", W3C Recommendation, April 2007, <http://www.w3.org/TR/2007/REC-soap12-part2-20070427/>

[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>

[X509] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks", Recommendation X.509, August 2005, <http://www.itu.int/rec/T-REC-X.509/en>

[X690] ITU-T, "Information Technology - ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", Recommendation X.690, July 2002, <http://www.itu.int/rec/T-REC-X.690/en>

[XMLNS-2ED] Bray, T., Hollander, D., Layman, A., and Tobin, R., Eds., "Namespaces in XML 1.0 (Second Edition)", W3C Recommendation, August 2006, <http://www.w3.org/TR/2006/REC-xml-names-20060816/>

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

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

[XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", W3C Recommendation 16 August 2006, edited in place 29 September 2006, <http://www.w3.org/TR/2006/REC-xml-20060816/>

1.2.2 Informative References

None.

1.3 Overview

The [MS-MWBF] specification defines a standard mechanism that can be used by a client to acquire a security token from a security token service (STS). Acquiring a security token is designed to address two problems related to communicating user information to remote applications and services.

First, in order to properly control access to information or resources in remote Web service (WS) resources, those WS resources have to have information about the users that are accessing them. Previous solutions required the WS resource to identify the user and use that identity to access further information about the user. Second, users were forced to be prompted multiple times to supply credentials (for example, user names and passwords) to securely identify themselves and authenticate to multiple WS resources.

Implementations of [MS-MWBF] solve these problems by moving the responsibility for authenticating the user away from the remote WS resource to an STS that already has an account for the user. This STS issues security tokens that contain information about the user in the form of claims. When accessing a WS resource, the user's web browser presents a security token obtained from an STS to the WS resource. The signature in the security token allows the WS resource to verify its validity, and the claims in the security token convey relevant user information to the WS resource. These claims can then be used for making authorization decisions by the WS resource.

In order for the protocol defined in [MS-MWBF] to work correctly, the WS resource has to obtain configuration information from the STS. This configuration information has to be obtained for the WS resource to send and receive [MS-MWBF] protocol messages. In addition to the information required prior to participating in [MS-MWBF], WS resources often need information about the STS configuration in order to enable WS resource users to understand the access control capabilities of the STS.

This specification defines a protocol that enables the WS resource to obtain the necessary information to configure the WS resource to participate in [MS-MWBF]. This protocol also allows for messages that enable the WS resource to obtain configuration information that is helpful for WS resource users, though not strictly necessary for participation in [MS-MWBF] exchanges.

The protocol is based on SOAP as defined in [SOAP1.1] and [SOAP1.2-1/2007]. The protocol defines the following operations:

- A GetFsTrustInformation operation that enables the WS resource to obtain configuration data from the STS that is necessary to participate in [MS-MWBF] exchanges, including URL endpoints, X.509 certificates for security token validation, and identifiers for the STS.
- A GetTrustedRealmUri operation that enables the WS resource to obtain configuration data from the STS that indicates to the security realms from which the STS accepts security tokens using [MS-MWBF].
- A GetClaims operation that enables the WS resource to obtain configuration data from the STS that indicates the security token claims that the STS can emit.

In section 3, the protocol specification describes the message processing model for the client and the STS to successfully emit or consume protocol messages that are created in accordance with section 2.

1.4 Relationship to Other Protocols

The Active Directory Federation Service (AD FS) Web Agent Protocol uses standard web protocols. To use this document effectively, the reader has to be familiar with the IETF specifications:

- Hypertext Transfer Protocol (HTTP), as specified in [RFC2616].
- Uniform Resource Identifiers (URIs), as specified in [RFC2396].
- Uniform Resource Locators (URLs), as specified in [RFC1738].

URLs and URIs are used to describe the data used in the protocol.

The Active Directory Federation Service (AD FS) Web Agent Protocol uses Extensible Markup Language (XML); the following specifications are used to describe the requirements for the XML syntax involved in the protocol. To use this document effectively, the reader has to be familiar with the following W3C specifications:

- Extensible Markup Language (XML) 1.0 (Fourth Edition), as specified in [XML].
- Namespaces in XML, as specified in [XMLNS-2ED].
- SOAP Version 1.1, as specified in [SOAP1.1].
- SOAP Version 1.2, as specified in [SOAP1.2-1/2007] and [SOAP1.2-2/2007].
- XML Schema Part 1: Structures Second Edition, as specified in [XMLSCHEMA1].
- XML Schema Part 2: Datatypes Second Edition, as specified in [XMLSCHEMA2].

1.5 Prerequisites/Preconditions

The client must be configured with the URL of the server's SOAP service in order to call the service.

1.6 Applicability Statement

The Active Directory Federation Service (AD FS) Web Agent Protocol is used by any software that needs knowledge of the configuration of an STS in order to validate security tokens from that STS. The software that needs knowledge of an STS's configuration is often WS resource software that expects to receive security tokens from users that are attempting to access the WS resource.

1.7 Versioning and Capability Negotiation

1.7.1 Versioning

This protocol uses the versioning mechanisms defined in the following specifications:

- SOAP 1.1, as specified in [SOAP1.1].
- SOAP 1.2, as specified in [SOAP1.2-1/2007] and [SOAP1.2-2/2007].

The data formatting and message processing of this protocol do not contain any further versioning mechanisms. The data itself is versioned to enable servers to determine whether clients need a full update or have an up-to-date version. This mechanism is described fully in sections 2 and 3 below.

1.7.2 Capability Negotiation

There is no functionality for capability negotiation in the Active Directory Federation Service (AD FS) Web Agent Protocol.

1.8 Vendor-Extensible Fields

As specified in section 2, the Active Directory Federation Service (AD FS) Web Agent Protocol uses SOAP messages for communication, as specified in [SOAP1.1] and [SOAP1.2-1/2007]. The core functionality of SOAP is to provide extensibility. [SOAP1.2-1/2007] and [SOAP1.1] contain detailed discussion on the SOAP messaging framework extensibility model. Vendors can use these SOAP extensibility points as specified.

1.9 Standards Assignments

There are no standards assignments for the Active Directory Federation Service (AD FS) Web Agent Protocol beyond those defined in the following specifications:

- SOAP 1.1, as specified in [SOAP1.1].
- SOAP 1.2, as specified in [SOAP1.2-1/2007] and [SOAP1.2-2/2007].

Unless otherwise indicated, Microsoft implementations follow all RECOMMENDED, SHOULD, MUST, MUST NOT, and SHOULD NOT prescriptions in the specifications listed above.

2 Messages

2.1 Transport

The GetFsTrustInformation request message (section 3.1.4.1.1.1), GetTrustedRealmUri request message (section 3.1.4.2.1.1), and GetClaims request message (section 3.1.4.3.1.1) MUST be transmitted using the HTTP POST method; they MUST NOT be transmitted using the GET method.

The client role and server role MUST use the HTTPS URL scheme to identify the server endpoints for processing GetFsTrustInformation request, GetTrustedRealmUri request, and GetClaims request messages.

2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses XML Schema (XSD) 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 using the mechanisms specified in [XMLNS-2ED]. 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
tns	http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/	
s1	http://microsoft.com/wsd/types/	
soap	http://schemas.xmlsoap.org/wsd/soap/	[WSDL]
mime	http://schemas.xmlsoap.org/wsd/mime/	[WSDL]
soap12	http://schemas.xmlsoap.org/wsd/soap12/	[WSDL]
http	http://schemas.xmlsoap.org/wsd/http/	[WSDL]
wSDL	http://schemas.xmlsoap.org/wsd/	[WSDL]
soap	http://schemas.xmlsoap.org/soap/envelope/	[SOAP1.1]
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
xsi	http://www.w3.org/2001/XMLSchema-instance	[XMLSCHEMA1]
xsd	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]

2.2.2 Messages

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

2.2.3 Elements

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

2.2.4 Complex Types

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

2.2.5 Simple Types

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

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3 Protocol Details

This section addresses the message processing model for the protocol. It includes related information required by an implementation to successfully emit or consume protocol messages, such as an abstract data model for maintaining configuration or state information.

The protocol specifies two distinct roles for the entities that emit and consume protocol messages. The server and the client roles are described in separate subsections below.

3.1 Server Details

This section describes details of protocol processing that must be understood in order to implement a server that can correctly perform its role in the protocol message exchange.

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The data used by each message exchange is different. The abstract data models for the GetFsTrustInformation, GetTrustedRealmUri, and GetClaims message exchanges can be found in the following sections.

3.1.1.1 GetFsTrustInformation

The client calls this method to get the information the client needs to verify security tokens issued by the server to the client using the protocol defined in [MS-MWBF]. The following data is used in the request and response.

Name	Description	Corresponding message parameter
Client Policy GUID	This is a globally unique identifier (GUID)* for the policy that is held by the client at the time of a GetFsTrustInformation request. If the client does not have this value (such as prior to emitting the first protocol request), it is required to represent it in the protocol as "00000000-0000-0000-0000-000000000000".	All Requests: Guid element
Server Policy GUID	This is a globally unique identifier for the policy that is maintained by the server at the time of issuing a GetFsTrustInformation response.	All Responses: Guid element
Client Policy Version	This is a version number for the policy that is held by the client at the time of a GetFsTrustInformation request. If the client does not have this value (such as prior to emitting the first protocol request), it is required to represent it in the protocol as "0".	All Requests: Version element
Server Policy Version	This is a version number for the policy that is maintained by the server at the time of issuing a GetFsTrustInformation response.	All Responses: Version element
Trusted Certificates	This is a list of identifiers for the certificates that can be used to sign security tokens targeted at the client. The identifiers are used to identify the certificates contained in the Federation Certificates data item discussed below.	All Responses: TrustedCertificates
Revocation Flags	This is a value that indicates whether and how revocation of X.509 certificates contained in the Federation Certificates list is to be checked.	All Responses: RevocationCheckFlags
Federation Certificates	This is a list of the X.509 certificates and their corresponding X.509 certificate issuer chains that can be used to sign security	All Responses: certificates

Name	Description	Corresponding message parameter
	tokens targeted at the client. The X.509 certificates in this collection that can be used to sign security tokens are identified by the Trusted Certificates data item described above.	
Federation Service Domain Account	This is a service principal name that identifies the domain account under which the server is running.	All Responses: fsDomainAccount
Hosted Realm URI	This is an identifier for the server. This URI is used in security tokens to identify the server as the issuer of the security token.	All Responses: hostedRealmUri
Login Service URL	This is the URL to which the client redirects service requests using the protocol described in [MS-MWBF].	All Responses: IsUrl

* Unless otherwise specified, all GUID values in this document follow the pattern specified for the "guid" simple type, which is first defined in section 3.1.4.1.1.1.

3.1.1.2 GetTrustedRealmUri

At the client, a higher layer might need to know whether the server accepts security tokens from a particular user's security realm as described in [MS-MWBF]. The user is represented by an email address. The client calls this method to determine whether the email address belongs to a security realm from which the server will accept tokens using the protocol defined in [MS-MWBF]. The following data is used in the request and response.

Name	Description	Corresponding message parameter
Email	This is an [RFC2822] email address. As noted above, the client calls GetTrustedRealmUri to find out whether the suffix of the email address represents a security realm from which the server is configured to accept tokens.	All Requests: email element
Trusted Realm URI	This is a URI that identifies the security realm corresponding to the submitted email suffix from which the server is configured to accept security tokens using [MS-MWBF].	All Responses: trustedRealmUri element

3.1.1.3 GetClaims

The client calls this method to discover what claims can be issued inside a security token using the protocol defined in [MS-MWBF]. The following data is used in the request and response.

Name	Description	Corresponding message parameter
Claim Type	This item identifies the type of claims that the client is discovering. This value MUST be Group.	Request: claimType element Response: groupClaimCollection
Custom Claim Name	Unused.	All Responses: Unused. This element MUST be empty.
Claim Value	This is the string value of the claim described.	All Responses: GroupClaim element

3.1.2 Timers

There are no protocol-specific timer events that must be serviced by an implementation. This protocol does not require timers beyond those that might be used by the underlying transport to transmit and

receive messages over HTTPS. The protocol does not include provisions for time-based retry for sending protocol messages.

3.1.3 Initialization

The initialization steps required for each of the three protocol message request and response pairs are unrelated to one another. Prior to sending any protocol message, the client **MUST** be configured with the URL to which the request is to be sent. The following sections define the initialization required for the client role prior to emitting each request message.

3.1.4 Message Processing Events and Sequencing Rules

The WSDL operations detailed in this section are unrelated to one another.

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

Operation	Description
GetFsTrustInformation	Enables the WS resource to obtain configuration data from the STS that is necessary to participate in the exchanges that are detailed in [MS-MWBF].
GetTrustedRealmUri	Enables the WS resource to obtain configuration data from the STS that indicates security realms.
GetClaims	Enables the WS resource to obtain configuration data from the STS that indicates security token claims.

Note All protocol messages **MUST** be well-formed XML placed within a SOAP envelope conforming to [SOAP1.2-1/2007] section 5.1 or [SOAP1.1] section 4. When the server receives a request that does not conform to the protocol, the server **MUST** return a SOAP fault.

3.1.4.1 GetFsTrustInformation

The GetFsTrustInformation exchange **MUST** consist of a single request message and a single response message. The exchange **MUST** be initiated by the client with a request message to the server. The following sections describe the server processing for the request and response messages.

3.1.4.1.1 Messages

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

Message	Description
GetFsTrustInformationSoapIn	The GetFsTrustInformation request message.
GetFsTrustInformationSoapOut	The GetFsTrustInformation response message.

3.1.4.1.1.1 GetFsTrustInformationSoapIn

The SOAP body of the GetFsTrustInformation request message **MUST** conform to the following XML schema.

```
<s:element name="GetFsTrustInformation">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="wsVersion" type="tns:VersionInformation"
    />
  </s:sequence>
</s:complexType>
```

```

</s:element>
<s:complexType name="VersionInformation">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="SoftwareVersion" type="s:long" />
    <s:element minOccurs="0" maxOccurs="1" name="Guid" type="sl:guid" />
    <s:element minOccurs="0" maxOccurs="1" name="Version" type="s:long" />
  </s:sequence>
</s:complexType>
<s:simpleType name="guid">
  <s:restriction base="s:string">
    <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}" />
  </s:restriction>
</s:simpleType>

```

3.1.4.1.1.1 SoftwareVersion Parameter

The value of this parameter MUST be 1. The server MUST ignore this value.

3.1.4.1.1.2 GetFsTrustInformationSoapIn Processing

If the request does not contain a wsVersion element, the server MUST proceed as if the client has an outdated copy; otherwise, the version number and GUID parameters in GetFsTrustInformation requests MUST be compared to the current version number and GUID of the server's local configuration. If the GUID in the request is different from the GUID of the server's local configuration, then the client has an outdated copy. If the version number in the request is less than the version number of the server's local configuration, then the client has an outdated copy. Otherwise, the client has an up-to-date copy. For the corresponding response processing, see section 3.1.4.1.1.4.

3.1.4.1.1.3 GetFsTrustInformationSoapOut

The SOAP body of the GetFsTrustInformation response message MUST conform to the following XML schema.

```

<s:element name="GetFsTrustInformationResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetFsTrustInformationResult"
type="s:boolean" />
      <s:element minOccurs="0" maxOccurs="1" name="fsVersion" type="tns:VersionInformation"
/>
      <s:element minOccurs="0" maxOccurs="1" name="trustInfo" type="tns:FsWithInformationData" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="VersionInformation">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="SoftwareVersion" type="s:long" />
    <s:element minOccurs="0" maxOccurs="1" name="Guid" type="sl:guid" />
    <s:element minOccurs="0" maxOccurs="1" name="Version" type="s:long" />
  </s:sequence>
</s:complexType>
<s:simpleType name="guid">
  <s:restriction base="s:string">
    <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}" />
  </s:restriction>
</s:simpleType>
<s:complexType name="FsWithInformationData">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="verificationMethod"
type="tns:X509VerificationMethod" />
  </s:sequence>
</s:complexType>

```

```

    <s:element minOccurs="1" maxOccurs="1" name="certificates"
type="tns:FederationCertificates" />
    <s:element minOccurs="0" maxOccurs="1" name="fsDomainAccount" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="hostedRealmUri" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="lsUrl" type="s:string" />
  </s:sequence>
</s:complexType>
<s:complexType name="X509VerificationMethod">
  <s:complexContent mixed="false">
    <s:extension base="tns:VerificationMethod">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="TrustedCertificates"
type="tns:ArrayOfCertInfo" />
        <s:element minOccurs="1" maxOccurs="1" name="RevocationCheckFlags"
type="tns:RevocationFlags" />
      </s:sequence>
    </s:extension>
  </s:complexContent>
</s:complexType>
<s:complexType name="VerificationMethod" abstract="true" />
<s:complexType name="ArrayOfCertInfo">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="CertInfo" nillable="true"
type="tns:CertInfo" />
  </s:sequence>
</s:complexType>
<s:complexType name="CertInfo">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="X509Thumbprint" type="s:string" />
  </s:sequence>
</s:complexType>
<s:simpleType name="RevocationFlags">
  <s:restriction base="s:string">
    <s:enumeration value="None" />
    <s:enumeration value="CheckEndCert" />
    <s:enumeration value="CheckEndCertCacheOnly" />
    <s:enumeration value="CheckChain" />
    <s:enumeration value="CheckChainCacheOnly" />
    <s:enumeration value="CheckChainExcludeRoot" />
    <s:enumeration value="CheckChainExcludeRootCacheOnly" />
  </s:restriction>
</s:simpleType>
<s:complexType name="FederationCertificates">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="SerializedStore" type="s:base64Binary" />
  </s:sequence>
</s:complexType>

```

3.1.4.1.1.3.1 SoftwareVersion Parameter

The value of this parameter MUST be 1.

3.1.4.1.1.3.2 hostedRealmUri Parameter

This parameter MUST be a URI conforming to [RFC2396].

3.1.4.1.1.3.3 lsUrl Parameter

This parameter MUST be a URL conforming to [RFC1738].

3.1.4.1.1.3.4 fsDomainAccount Parameter

This parameter MUST be a NetBIOS name, followed by a backslash ("\"), and finally followed by a SAMAccountName as described in [MS-ADTS] section 3.1.1.3.2.38.

3.1.4.1.1.3.5 X509Thumbprint Parameter

This parameter MUST contain a sequence of 40 hexadecimal digital characters. The characters MUST be limited to alphanumeric hexadecimal digits. If the character is an alphabetical character, it MUST be uppercased. Specifically, the characters MUST only be 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, or F. The sequence MUST NOT contain any spaces.

3.1.4.1.1.3.6 SerializedStore Parameter

This parameter MUST contain an encoded list of certificates. The format of this list MUST be a CMS signed-data content type as defined by [RFC3852] section 5.1. The list MUST be initially encoded using the Distinguished Encoding Rules (DER) as defined by [X690]. The DER-encoded lists MUST then be base 64-encoded per [RFC4648].

According to [RFC3852] section 3, the basic CMS structure consists of the following.

```
ContentInfo ::= SEQUENCE {
    contentType ContentType,
    content [0] EXPLICIT ANY DEFINED BY contentType }
ContentType ::= OBJECT IDENTIFIER
```

As described in [RFC3852] section 5.1, the **contentType** field MUST be set to the object identifier (OID) value of "1.2.840.113549.1.7.2". As described in [RFC3852] section 5.1, the **content** field is a SignedData type that has the following structure.

```
SignedData ::= SEQUENCE {
    version CMSVersion,
    digestAlgorithms DigestAlgorithmIdentifiers,
    encapContentInfo EncapsulatedContentInfo,
    certificates [0] IMPLICIT CertificateSet OPTIONAL,
    crls [1] IMPLICIT RevocationInfoChoices OPTIONAL,
    signerInfos SignerInfos }
DigestAlgorithmIdentifiers ::= SET OF DigestAlgorithmIdentifier
SignerInfos ::= SET OF SignerInfo
```

The **version** field MUST equal 1 according to the description of the **version** field on page 8 and 9 of [RFC3852]. The **digestAlgorithms** field MUST be an empty set. The **encapContentInfo** field consists of the following structure according to [RFC3852] section 5.2.

```
EncapsulatedContentInfo ::= SEQUENCE {
    eContentType ContentType,
    eContent [0] EXPLICIT OCTET STRING OPTIONAL }
ContentType ::= OBJECT IDENTIFIER
```

The **eContentType** field MUST equal the value "1.2.840.113549.1.7.1". The **eContent** field MUST be null.

The **certificates** field of the SignedData sequence MUST contain a list of X.509 certificates conforming to [X509]. The **crls** field MUST be null. The **SignerInfos** field MUST be empty.

3.1.4.1.1.4 GetFsTrustInformationSoapOut Processing

Response processing can be divided into version processing, certificates processing, and other processing. The following sections address this processing.

3.1.4.1.1.4.1 Versioning Processing

If the client's data is up-to-date, as described in section 3.1.4.1.1.2, then the `GetFsTrustInformationResult` MUST be set to false, and the `VersionInformation` and `FsInformation` elements described in section 3.1.4.1.1.3 MUST be omitted from the response.

If the client's data is an outdated copy, then the `GetFsTrustInformationResult` MUST be set to true, and the `VersionInformation` and `FsInformation` elements described in section 3.1.4.1.1.3 MUST be included in the response.

The `Version` element MUST be set to the version number for the current configuration maintained by the server. The `Guid` element MUST be set to the GUID for the current configuration maintained by the server.

3.1.4.1.1.4.2 Certificate Processing

The server MUST maintain a list of at least one X.509 certificate that is used for signing security tokens. For each of the X.509 certificates, the server MUST maintain a SHA-1 hash of the certificate.

When responding to a `GetFsInformation` request, the server MUST place each of the X.509 certificate hashes into an `X509Thumbprint` element as described in section 3.1.4.1.1.3.5. The hash data MUST be represented as a sequence of sets of two hexadecimal digit characters. There MUST NOT be any spaces between the characters.

When responding to a `GetFsInformation` request, the server MUST place each of the maintained X.509 certificates into the data structure described in section 3.1.4.1.1.3.6. In addition, the server MUST put all of the X.509 certificates that are in the issuance path of the X.509 certificates used for signing security tokens into the data structure described in section 3.1.4.1.1.3.6. [X509] discusses how to determine the other X.509 certificates that are in the issuance path of given X.509 certificate.

The server MUST maintain a configured method for checking revocation on the X.509 certificates as described in section 3.2.4.1.2.2. The method MUST be included in the response `RevocationFlags` element, as detailed in section 3.1.4.1.1.3.

3.1.4.1.1.4.3 Other Processing

The server MUST maintain a URI to identify itself as described in section 3.1.1.1. This URI MUST be included in the response as the `hostedRealmUri` element.

The server MUST maintain a URL that represents the endpoint on which it listens for [MS-MWBF] requests. This URL MUST be included in the response as the `IsUrl` element.

Finally, the server MUST include the service principal name of the domain account under which it runs according to [MS-ADTS] as the `fsDomainAccount` element.

3.1.4.2 GetTrustedRealmUri

The `GetTrustedRealmUri` exchange consists of a single request message and a single response message. The exchange MUST be initiated by the client with a request message to the server. The following sections describe the server processing for the request and response messages.

3.1.4.2.1 Messages

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

Message	Description
<code>GetTrustedRealmUriSoapIn</code>	The <code>GetTrustedRealmUri</code> request message.

Message	Description
GetTrustedRealmUriSoapOut	The GetTrustedRealmUri response message.

3.1.4.2.1.1 GetTrustedRealmUriSoapIn

The SOAP body of the GetTrustedRealmUri request message MUST conform to the following XML schema.

```
<s:element name="GetTrustedRealmUri">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="email" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

3.1.4.2.1.1.1 Email Parameter

The element named "email" MUST contain an email address conforming to [RFC2822].

3.1.4.2.1.2 GetTrustedRealmUriSoapIn Processing

As described in section 3.2.3.2, the client MUST include an email address in the request.

3.1.4.2.1.3 GetTrustedRealmUriSoapOut

The SOAP body of the GetTrustedRealmUri response message MUST conform to the following XML schema.

```
<s:element name="GetTrustedRealmUriResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetTrustedRealmUriResult" type="s:boolean" />
      <s:element minOccurs="0" maxOccurs="1" name="trustedRealmUri" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

3.1.4.2.1.3.1 trustedRealmUri

The "trustedRealmUri" element MUST contain a URI conforming to [RFC2396].

3.1.4.2.1.4 GetTrustedRealmUriSoapOut Processing

If the suffix of the [RFC2822] email address from the request matches the local configuration of the server for a security realm from which the server accepts security tokens OR if there exists a trusted account store that the server uses to authenticate users, then the GetTrustedRealmUriResult element MUST be true. Otherwise, the GetTrustedRealmUriResult element MUST be false. If the GetTrustedRealmUriResult element is true, then the trustedRealmUri element MUST include the URI maintained by the server to identify the security realm. If the GetTrustedRealmUriResult element is false, then the trustedRealmUri element MUST be omitted.

3.1.4.3 GetClaims

The GetClaims exchange consists of a single request message and a single response message. The exchange MUST be initiated by the client with a request message to the server. The following sections describe the server processing for the request and response messages.

3.1.4.3.1 Messages

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

Message	Description
GetClaimsSoapIn	The GetClaims request message.
GetClaimsSoapOut	The GetClaims response message.

3.1.4.3.1.1 GetClaimsSoapIn

The SOAP body of the GetClaims request message MUST conform to the following XML schema.

```
<s:element name="GetClaims">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="claimType" type="tns:ClaimType" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:simpleType name="ClaimType">
  <s:restriction base="s:string">
    <s:enumeration value="Group" />
    <s:enumeration value="Custom" />
    <s:enumeration value="GroupAndCustom" />
  </s:restriction>
</s:simpleType>
```

3.1.4.3.1.2 GetClaimsSoapIn Processing

As described in section 3.2.3.3, the client MUST set the claim type in the request to "Group".

3.1.4.3.1.3 GetClaimsSoapOut

The SOAP body of the GetClaims response message MUST conform to the following XML schema.

```
<s:element name="GetClaimsResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="groupClaimCollection"
type="tns:ArrayOfGroupClaim" />
      <s:element minOccurs="0" maxOccurs="1" name="customClaimCollection"
type="tns:ArrayOfCustomClaim" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="ArrayOfGroupClaim">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="GroupClaim" nillable="true"
type="tns:GroupClaim" />
  </s:sequence>
</s:complexType>
<s:complexType name="GroupClaim" mixed="true">
```

```

    <s:complexContent mixed="false">
      <s:extension base="tns:TrustPolicyEntryBase">
        <s:attribute name="IsSensitive" type="s:boolean" use="required" />
      </s:extension>
    </s:complexContent>
  </s:complexType>
</s:complexType name="TrustPolicyEntryBase">
  <s:attribute name="uuid" type="s1:guid" use="required" />
  <s:attribute name="Disabled" type="s:boolean" use="required" />
</s:complexType>
<s:complexType name="CustomClaim">
  <s:complexContent mixed="false">
    <s:extension base="tns:TrustPolicyEntryBase">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="CustomClaimName" type="s:string" />
      </s:sequence>
      <s:attribute name="IsSensitive" type="s:boolean" use="required" />
    </s:extension>
  </s:complexContent>
</s:complexType>
<s:complexType name="ActiveDirectoryGroupClaim">
  <s:complexContent mixed="true">
    <s:extension base="tns:GroupClaim">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="GroupSid" type="s:string" />
      </s:sequence>
    </s:extension>
  </s:complexContent>
</s:complexType>
<s:complexType name="ArrayOfCustomClaim">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="CustomClaim" nillable="true"
type="tns:CustomClaim" />
  </s:sequence>
</s:complexType>
<s:simpleType name="guid">
  <s:restriction base="s:string">
    <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}" />
  </s:restriction>
</s:simpleType>

```

3.1.4.3.1.4 GetClaimsSoapOut Processing

The collection of claims that is returned MUST represent the group claims that can appear in a security token from the server.

3.1.5 Timer Events

There are no protocol-specific timer events that must be serviced by an implementation. This protocol does not require timers beyond those that might be used by the underlying transport to transmit and receive messages over HTTPS. The protocol does not include provisions for time-based retry for sending protocol messages.

3.1.6 Other Local Events

This protocol does not have dependencies on any transport protocols other than HTTP 1.1. This protocol relies on this transport mechanism for the correct and timely delivery of protocol messages. The protocol does not take action in response to any changes or failure in machine state or network communications.

3.2 Client Details

This section describes details of protocol processing that must be understood to implement a client that can correctly perform its role in the protocol message exchange.

3.2.1 Abstract Data Model

The abstract data model described in section 3.1.1 applies for the client role as well.

3.2.2 Timers

There are no protocol-specific timer events that must be serviced by an implementation. This protocol does not require timers beyond those that might be used by the underlying transport to transmit and receive messages over HTTPS. The protocol does not include provisions for time-based retry for sending protocol messages.

3.2.3 Initialization

Prior to receiving request messages, the server MUST open an endpoint to listen for request messages. In order to provide the data described in the abstract data model, that data MUST be configured on the server by an administrator.

3.2.3.1 GetFsTrustInformation Initialization

The client can maintain a cached copy of the data described in section 3.1.1.1.<1>

Prior to emitting a GetFsTrustInformation request, the client MUST obtain the version number and GUID of the currently cached trust information. If no trust information is cached on the client, the client MUST use a version number equal to 0 and a GUID equal to 00000000-0000-0000-0000-000000000000.

3.2.3.2 GetTrustedRealmUri Initialization

Prior to emitting a GetTrustedRealmUri request, the client MUST have an email address conforming to [RFC2822] to include in the request message. This information MUST be provided by a higher layer, as described in section 3.2.4.2.

3.2.3.3 GetClaims Initialization

Prior to emitting a GetClaims request, the client MUST set the claim type to "Group" for the request.

3.2.4 Message Processing Events and Sequencing Rules

The WSDL operations detailed in section 3.1.4 are unrelated to one another. A client MUST emit request messages according to the events that trigger the requests as described in the following sections. The following sections define the message processing rules separately for the GetFsTrustInformation, GetTrustedRealmUri, and GetClaims message exchanges.

3.2.4.1 GetFsTrustInformation

The GetFsTrustInformation exchange MUST consist of a single request message and a single response message. The exchange MUST be initiated by the client with a request message to the server.

As described in section 3.1.1.1, the client emits a GetFsTrustInformation request when the client needs the data described in section 3.1.1.1 to verify the security tokens issued by the server. Thus, a

GetFsTrustInformation request MAY be triggered by the receipt of a security token from the server. Implementations MAY choose to improve the performance of security token verification by emitting a GetFsTrustInformation request and caching the data from the response prior to receiving a security token from the server.<2>

The following sections describe the client processing for the request and response messages.

3.2.4.1.1 GetFsTrustInformationSoapIn Processing

As described in section 3.2.3.1, the client MUST include the current policy version number and corresponding GUID in the request.

3.2.4.1.2 GetFsTrustInformationSoapOut Processing

GetFsTrustInformation response processing can be divided into version processing, certificates processing, and other processing. The following sections discuss these processing steps.

3.2.4.1.2.1 Versioning

As detailed in section 3.1.4.1.1.3, the response MUST contain a version number and GUID representing the configuration data described in section 3.1.1.1. This version number and GUID MUST be compared to the locally cached information. If the GUID from the response is different from the GUID cached locally, then the response contains newer data that MUST be used instead of the locally cached data. If the response GUID and locally cached GUID are identical, but the locally cached version number is smaller than the response version number, then the response contains newer data that MUST be used instead of the locally cached data. If there is no locally cached data, the version number and GUID SHOULD be cached for use in processing future server responses.

3.2.4.1.2.2 Certificates

The abstract data model for the certificates is described in section 3.2.3.1. The Federation Certificates data described in section 3.2.3.1 MUST include at least one X.509 certificate. The Trusted Certificates data described in section 3.2.3.1 MUST include at least one identifier that matches an X.509 certificate found in the Federation Certificates data. Each X.509 certificate from Federation Certificates that is identified in Trusted Certificates is stored locally as a way to verify security token signatures from the server. [MS-MWBF] (section 5.1.2) describes a method to perform the verification.

The Revocation Flags data described in section 3.2.3.1 is stored locally. The semantics of this data are described in the following table.

Value	Description
CheckChain	Revocation checking MUST be done on all of the certificates in every chain.
CheckChainCacheOnly	Revocation checking MUST be done on all of the certificates in every chain. Revocation checking MUST only use locally cached revocation data.
CheckChainExcludeRoot	Revocation checking MUST be done on all certificates in all of the chains except the root certificate of each chain.
CheckChainExcludeRootCacheOnly	Revocation checking MUST be done on all certificates in all of the chains except the root certificate of each chain. Revocation checking MUST only use locally cached revocation data.
CheckEndCert	Revocation checking is done on the end certificate and only the end certificate.
CheckEndCertCacheOnly	Revocation checking is done on the end certificate and only the end certificate. Revocation checking MUST only use locally cached revocation data.
None	Revocation checking MUST NOT be done.

3.2.4.1.2.3 Other Data

The other data contained in the response MUST be cached for use in the protocol described in [MS-MWBF]. The Login Service URL MUST be cached for the purpose of redirecting requests according to [MS-MWBF]. The Hosted Realm URI MUST be cached for the purpose of identifying the server and verifying the **issuer** field in received security tokens. The Federation Service Domain Account name MUST be cached for the purpose of verifying incoming security token signatures.

3.2.4.2 GetTrustedRealmUri

The GetTrustedRealmUri exchange consists of a single request message and a single response message. The exchange MUST be initiated by the client with a request message to the server.

As described in section 3.1.1.2, a GetTrustedRealmUri request is emitted when a higher layer needs to check whether the server accepts security tokens from the security realm indicated by the suffix of an email address. Thus higher layers, such as users of software that call into the local client APIs, will trigger the emission of a GetTrustedRealmUri request.

The following sections describe the client processing for the request and response messages.

3.2.4.2.1 GetTrustedRealmUriSoapIn Processing

Upon receiving the GetTrustedRealmUri request, the email specified in the request MUST be extracted for use in processing the response. There is no other request-specific processing.

3.2.4.2.2 GetTrustedRealmUriSoapOut Processing

The GetTrustedRealmUriResult response element described in section 3.1.4.2.1.3 MUST indicate whether or not the email address matches a security realm from which the server accepts security tokens and MUST contain the trustedRealmUri element as described in section 3.1.4.2.1.4.

The result and the security realm URI MUST be provided to the higher layer that requested the information.

3.2.4.3 GetClaims

The GetClaims exchange consists of a single request message and a single response message. The exchange MUST be initiated by the client with a request message to the server.

As described in section 3.1.1.3, a GetClaims request is emitted when a higher layer needs the list of group claims that can be emitted by the server. Thus higher layers, such as users of software that call into the local client APIs, will trigger the emission of a GetClaims request.

The following sections describe the client processing for the request and response messages.

3.2.4.3.1 GetClaimsSoapIn Processing

Upon receiving the GetClaims request, the claim type specified in the request MUST be extracted for use in processing the response. There is no other request-specific processing.

3.2.4.3.2 GetClaimsSoapOut Processing

The server MUST return a groupClaimCollection that contains each of the configured group claim values.

3.2.5 Timer Events

There are no protocol-specific timer events that must be serviced by an implementation. This protocol does not require timers beyond those that might be used by the underlying transport to transmit and receive messages over HTTPS. The protocol does not include provisions for time-based retry for sending protocol messages.

3.2.6 Other Local Events

This protocol does not have dependencies on any transport protocols other than HTTP 1.1. This protocol relies on this transport mechanism for the correct and timely delivery of protocol messages. The protocol does not take action in response to any changes or failure in machine state or network communications.

4 Protocol Examples

The examples shown in subsequent sections are based on the full WSDL defined in section 6.

4.1 GetFsTrustInformation Request Message Example

```
<?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>
      <GetFsTrustInformation
        xmlns="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/"
        <wsVersion>
          <SoftwareVersion>
            1
          </SoftwareVersion>
          <Guid>
            00000000-0000-0000-0000-000000000000
          </Guid>
          <Version>
            0
          </Version>
        </wsVersion>
      </GetFsTrustInformation>
    </soap:Body>
  </soap:Envelope>
```

4.2 GetFsTrustInformation Response Message Example

```
<?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>
      <GetFsTrustInformationResponse
        xmlns="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/"
        <GetFsTrustInformationResult>
          true
        </GetFsTrustInformationResult>
        <fsVersion>
          <SoftwareVersion>
            1
          </SoftwareVersion>
          <Guid>
            c8fbb077-6f57-43b9-a8c1-1884fe8813b5
          </Guid>
          <Version>
            16
          </Version>
        </fsVersion>
        <trustInfo>
          <verificationMethod>
            <TrustedCertificates>
              <CertInfo>
                <X509Thumbprint>
                  2439425F275E7981AA79895BF13CEFD6B026F0D6
                </X509Thumbprint>
              </CertInfo>
              <CertInfo>
                <X509Thumbprint>
                  17F649094B9C28F3ED3ADEA90CAD485474ED017C
                </X509Thumbprint>
              </CertInfo>
            </TrustedCertificates>
            <RevocationCheckFlags>
```

```

        CheckChainExcludeRoot
      </RevocationCheckFlags>
    </verificationMethod>
    <certificates>
      <SerializedStore>
MIIIRwYJKoZIhvcNAQcCoIIODCCJTQCAQExADALBqkqhkiG9w0BBwGggiUcMIIIFcDCCBNmgAwIBAgIQigCG3AnlJ5VC6
K7trUKusjANBgkqhkiG9w0BAQUFADCbDETMBEgcGmSJomT8ixkARkWA2NvbTEZMBcGCmSJomT8ixkARkWCWlpY3Jvc2
9mdDEWMBQGCmSJomT8ixkARkWBm50dGVzdDEcMBoGCmSJomT8ixkARkWDGFkZnNhZG9tbGg...
      </SerializedStore>
    </certificates>
    <fsDomainAccount>
      ADFSRDOMLH-2\DSP20A46$
    </fsDomainAccount>
    <hostedRealmUri>
      urn:federation:trey research
    </hostedRealmUri>
    <lsUrl>https://DSP20A46.adfsrdomlh-2.nttest.microsoft.com/adfs/ls/</lsUrl>
  </trustInfo>
</GetFsTrustInformationResponse>
</soap:Body>
</soap:Envelope>

```

4.3 GetTrustedRealmUri Request Message Example

```

<?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>
      <GetTrustedRealmUri
        xmlns="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/">
        <email>
          user@adatum.com
        </email>
      </GetTrustedRealmUri>
    </soap:Body>
  </soap:Envelope>

```

4.4 GetTrustedRealmUri Response Message Example

```

<?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>
      <GetTrustedRealmUriResponse
        xmlns="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/">
        <GetTrustedRealmUriResult>
          true
        </GetTrustedRealmUriResult>
        <trustedRealmUri>
          urn:federation:trey research
        </trustedRealmUri>
      </GetTrustedRealmUriResponse>
    </soap:Body>
  </soap:Envelope>

```

4.5 GetClaims Request Message Example

```

<?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>
        <GetClaims
xmlns="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/">
            <claimType>
                Group
            </claimType>
        </GetClaims>
    </soap:Body>
</soap:Envelope>

```

4.6 GetClaims Response Message Example

```

<?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>
        <GetClaimsResponse
xmlns="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/">
            <groupClaimCollection>
                <GroupClaim xsi:type="ActiveDirectoryGroupClaim" uuid="13f634f2-047b-4f31-a0a4-
37e47770ab8c" Disabled="false" IsSensitive="false">
                    Form Approver
                    <GroupSid>S-1-5-21-364685705-101228684-3468600125-101099</GroupSid>
                </GroupClaim>
            </groupClaimCollection>
        </GetClaimsResponse>
    </soap:Body>
</soap:Envelope>

```

5 Security

5.1 Security Considerations for Implementers

Implementers have to ensure that SSL is used to authenticate that the server is the intended server referred to by the server endpoint URL. Otherwise, there are no specific security considerations beyond those specified in normative references.

5.2 Index of Security Parameters

None of the protocol parameters are specific to the security of the protocol.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided below:

```
<?xml version="1.0" encoding="utf-8"?>
<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://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/"
xmlns:sl="http://microsoft.com/wsdl/types/" 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://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/">
      <s:import namespace="http://microsoft.com/wsdl/types/" />
      <s:element name="LsRequestSecurityToken">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="credentialTypeUri" type="s:string"
/>
            <s:element minOccurs="0" maxOccurs="1" name="credentials"
type="tns:ArrayOfString" />
            <s:element minOccurs="0" maxOccurs="1" name="accountStoreUri" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="cookie" type="s:base64Binary" />
            <s:element minOccurs="0" maxOccurs="1" name="targetRealmName" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfString">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="LsRequestSecurityTokenResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="rstr" type="tns:RSTRResult" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="RSTRResult">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="Status" type="tns:RSTRStatus" />
          <s:element minOccurs="0" maxOccurs="1" name="PolicyVersion"
type="tns:VersionInformation" />
          <s:element minOccurs="0" maxOccurs="1" name="CredentialsVerification"
type="tns:CredentialsVerificationInfo" />
          <s:element minOccurs="0" maxOccurs="1" name="ForeignRealmUri" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="SecurityToken" type="s:base64Binary"
/>
          <s:element minOccurs="0" maxOccurs="1" name="LogonAcceleratorToken"
type="s:base64Binary" />
        </s:sequence>
      </s:complexType>
      <s:simpleType name="RSTRStatus">
        <s:restriction base="s:string">
          <s:enumeration value="Success" />
          <s:enumeration value="WrongPrincipal" />
          <s:enumeration value="NoAcceptableCredential" />
          <s:enumeration value="InvalidTarget" />
          <s:enumeration value="ValidationFailure" />
          <s:enumeration value="GenerationFailure" />
          <s:enumeration value="SidExpansionFailure" />
        </s:restriction>
      </s:simpleType>
    </s:schema>
  </wsdl:types>

```

```

        <s:enumeration value="NoAccountStores" />
        <s:enumeration value="NoActiveDirectoryForSids" />
        <s:enumeration value="NoAccountStoresForCert" />
        <s:enumeration value="Unset" />
    </s:restriction>
</s:simpleType>
<s:complexType name="VersionInformation">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="SoftwareVersion" type="s:long" />
        <s:element minOccurs="0" maxOccurs="1" name="Guid" type="s1:guid" />
        <s:element minOccurs="0" maxOccurs="1" name="Version" type="s:long" />
    </s:sequence>
</s:complexType>
<s:complexType name="CredentialsVerificationInfo">
    <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="AccountStoreType"
type="tns:AccountStoreType" />
        <s:element minOccurs="0" maxOccurs="1" name="AccountStoreTypeDisplay"
type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="AccountStoreUriString" type="s:string"
/>
        <s:element minOccurs="0" maxOccurs="1" name="AccountStoreDisplayName"
type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="UserValidationData"
type="tns:UserValidationInfo" />
    </s:sequence>
</s:complexType>
<s:simpleType name="AccountStoreType">
    <s:restriction base="s:string">
        <s:enumeration value="ActiveDirectoryType" />
        <s:enumeration value="LdapDirectoryType" />
        <s:enumeration value="UnknownStoreType" />
    </s:restriction>
</s:simpleType>
<s:complexType name="UserValidationInfo">
    <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="ErrorCode" type="s:long" />
        <s:element minOccurs="0" maxOccurs="1" name="AdditionalValidationInfo"
type="tns:ArrayOfString" />
    </s:sequence>
</s:complexType>
<s:element name="RequestSecurityTokenWithToken">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="inToken" type="s:base64Binary" />
            <s:element minOccurs="0" maxOccurs="1" name="cookie" type="s:base64Binary" />
            <s:element minOccurs="0" maxOccurs="1" name="targetRealmName" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="RequestSecurityTokenWithTokenResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="rstr" type="tns:RSTRResult" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="LsRequestSecurityTokenWithCookie">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="latToken" type="s:base64Binary" />
            <s:element minOccurs="0" maxOccurs="1" name="targetRealmName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="authMethodUri"
type="tns:ArrayOfString" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="LsRequestSecurityTokenWithCookieResponse">
    <s:complexType>

```

```

        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="rstr" type="tns:RSTRResult" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetProxyTrustConfiguration">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="proxyVersion"
type="tns:VersionInformation" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="GetProxyTrustConfigurationResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="GetProxyTrustConfigurationResult"
type="s:boolean" />
          <s:element minOccurs="0" maxOccurs="1" name="fsVersion"
type="tns:VersionInformation" />
          <s:element minOccurs="0" maxOccurs="1" name="proxyInformation"
type="tns:ProxyInformation" />
          <s:element minOccurs="0" maxOccurs="1" name="trustConfig"
type="tns:ArrayOfTrustConfigurationData" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="ProxyInformation">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="HostedRealmUriStr" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="LsUrlStr" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="ConfigInfo"
type="tns:ProxyConfigurationInformation" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="ProxyConfigurationInformation">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="CookiePath" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="SuppressRealmCookie" type="s:boolean"
/>
        <s:element minOccurs="1" maxOccurs="1" name="RealmCookieLifetime" type="s:int" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="ArrayOfTrustConfigurationData">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="TrustConfigurationData"
nillable="true" type="tns:TrustConfigurationData" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="TrustConfigurationData">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="trustType" type="tns:TrustTypes" />
        <s:element minOccurs="0" maxOccurs="1" name="trustDisplayName" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="trustUri" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="trustLsUrl" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="acceptableAuthenticationMethodStrings"
type="tns:ArrayOfString" />
      </s:sequence>
    </s:complexType>
    <s:simpleType name="TrustTypes">
      <s:restriction base="s:string">
        <s:enumeration value="TrustedRealm" />
        <s:enumeration value="TrustingRealm" />
        <s:enumeration value="TrustingResource" />
        <s:enumeration value="SelfhostedRealm" />
        <s:enumeration value="UnknownTrustType" />
      </s:restriction>
    </s:simpleType>
  </s:element name="GetFsTrustInformation">

```

```

        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="wsVersion"
type="tns:VersionInformation" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetFsTrustInformationResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetFsTrustInformationResult"
type="s:boolean" />
            <s:element minOccurs="0" maxOccurs="1" name="fsVersion"
type="tns:VersionInformation" />
            <s:element minOccurs="0" maxOccurs="1" name="trustInfo"
type="tns:FsWithInformationData" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="FsWithInformationData">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="verificationMethod"
type="tns:X509VerificationMethod" />
          <s:element minOccurs="1" maxOccurs="1" name="certificates"
type="tns:FederationCertificates" />
          <s:element minOccurs="0" maxOccurs="1" name="fsDomainAccount" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="hostedRealmUri" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="lsUrl" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="X509VerificationMethod">
        <s:complexContent mixed="false">
          <s:extension base="tns:VerificationMethod">
            <s:sequence>
              <s:element minOccurs="0" maxOccurs="1" name="TrustedCertificates"
type="tns:ArrayOfCertInfo" />
              <s:element minOccurs="1" maxOccurs="1" name="RevocationCheckFlags"
type="tns:RevocationFlags" />
            </s:sequence>
          </s:extension>
        </s:complexContent>
      </s:complexType>
      <s:complexType name="VerificationMethod" abstract="true" />
      <s:complexType name="ArrayOfCertInfo">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="CertInfo" nillable="true"
type="tns:CertInfo" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="CertInfo">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="X509Thumbprint" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:simpleType name="RevocationFlags">
        <s:restriction base="s:string">
          <s:enumeration value="None" />
          <s:enumeration value="CheckEndCert" />
          <s:enumeration value="CheckEndCertCacheOnly" />
          <s:enumeration value="CheckChain" />
          <s:enumeration value="CheckChainCacheOnly" />
          <s:enumeration value="CheckChainExcludeRoot" />
          <s:enumeration value="CheckChainExcludeRootCacheOnly" />
        </s:restriction>
      </s:simpleType>
      <s:complexType name="FederationCertificates">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="SerializedStore" type="s:base64Binary"
/>>

```

```

    </s:sequence>
  </s:complexType>
  <s:element name="GetTrustedRealmUri">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="email" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetTrustedRealmUriResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="GetTrustedRealmUriResult"
type="s:boolean" />
        <s:element minOccurs="0" maxOccurs="1" name="trustedRealmUri" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="GetClaims">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="claimType" type="tns:ClaimType" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:simpleType name="ClaimType">
    <s:restriction base="s:string">
      <s:enumeration value="Group" />
      <s:enumeration value="Custom" />
      <s:enumeration value="GroupAndCustom" />
    </s:restriction>
  </s:simpleType>
  <s:element name="GetClaimsResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="groupClaimCollection"
type="tns:ArrayOfGroupClaim" />
        <s:element minOccurs="0" maxOccurs="1" name="customClaimCollection"
type="tns:ArrayOfCustomClaim" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:complexType name="ArrayOfGroupClaim">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="unbounded" name="GroupClaim" nillable="true"
type="tns:GroupClaim" />
    </s:sequence>
  </s:complexType>
  <s:complexType name="GroupClaim" mixed="true">
    <s:complexContent mixed="false">
      <s:extension base="tns:TrustPolicyEntryBase">
        <s:attribute name="IsSensitive" type="s:boolean" use="required" />
      </s:extension>
    </s:complexContent>
  </s:complexType>
  <s:complexType name="TrustPolicyEntryBase">
    <s:attribute name="uuid" type="s1:guid" use="required" />
    <s:attribute name="Disabled" type="s:boolean" use="required" />
  </s:complexType>
  <s:complexType name="CustomClaim">
    <s:complexContent mixed="false">
      <s:extension base="tns:TrustPolicyEntryBase">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="CustomClaimName" type="s:string"
/>
        </s:sequence>
        <s:attribute name="IsSensitive" type="s:boolean" use="required" />
      </s:extension>
    </s:complexContent>
  </s:complexType>

```

```

</s:complexType>
<s:complexType name="ActiveDirectoryGroupClaim">
  <s:complexContent mixed="true">
    <s:extension base="tns:GroupClaim">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="GroupSid" type="s:string" />
      </s:sequence>
    </s:extension>
  </s:complexContent>
</s:complexType>
<s:complexType name="ArrayOfCustomClaim">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="CustomClaim" nillable="true"
type="tns:CustomClaim" />
  </s:sequence>
</s:complexType>
</s:schema>
<s:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/wsdl/types/">
  <s:simpleType name="guid">
    <s:restriction base="s:string">
      <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-
9a-fA-F]{12}" />
    </s:restriction>
  </s:simpleType>
</s:schema>
</wsdl:types>
<wsdl:message name="LsRequestSecurityTokenSoapIn">
  <wsdl:part name="parameters" element="tns:LsRequestSecurityToken" />
</wsdl:message>
<wsdl:message name="LsRequestSecurityTokenSoapOut">
  <wsdl:part name="parameters" element="tns:LsRequestSecurityTokenResponse" />
</wsdl:message>
<wsdl:message name="RequestSecurityTokenWithTokenSoapIn">
  <wsdl:part name="parameters" element="tns:RequestSecurityTokenWithToken" />
</wsdl:message>
<wsdl:message name="RequestSecurityTokenWithTokenSoapOut">
  <wsdl:part name="parameters" element="tns:RequestSecurityTokenWithTokenResponse" />
</wsdl:message>
<wsdl:message name="LsRequestSecurityTokenWithCookieSoapIn">
  <wsdl:part name="parameters" element="tns:LsRequestSecurityTokenWithCookie" />
</wsdl:message>
<wsdl:message name="LsRequestSecurityTokenWithCookieSoapOut">
  <wsdl:part name="parameters" element="tns:LsRequestSecurityTokenWithCookieResponse" />
</wsdl:message>
<wsdl:message name="GetProxyTrustConfigurationSoapIn">
  <wsdl:part name="parameters" element="tns:GetProxyTrustConfiguration" />
</wsdl:message>
<wsdl:message name="GetProxyTrustConfigurationSoapOut">
  <wsdl:part name="parameters" element="tns:GetProxyTrustConfigurationResponse" />
</wsdl:message>
<wsdl:message name="GetFsTrustInformationSoapIn">
  <wsdl:part name="parameters" element="tns:GetFsTrustInformation" />
</wsdl:message>
<wsdl:message name="GetFsTrustInformationSoapOut">
  <wsdl:part name="parameters" element="tns:GetFsTrustInformationResponse" />
</wsdl:message>
<wsdl:message name="GetTrustedRealmUriSoapIn">
  <wsdl:part name="parameters" element="tns:GetTrustedRealmUri" />
</wsdl:message>
<wsdl:message name="GetTrustedRealmUriSoapOut">
  <wsdl:part name="parameters" element="tns:GetTrustedRealmUriResponse" />
</wsdl:message>
<wsdl:message name="GetClaimsSoapIn">
  <wsdl:part name="parameters" element="tns:GetClaims" />
</wsdl:message>
<wsdl:message name="GetClaimsSoapOut">
  <wsdl:part name="parameters" element="tns:GetClaimsResponse" />
</wsdl:message>

```

```

<wsdl:portType name="FederationServerServiceSoap">
  <wsdl:operation name="LsRequestSecurityToken">
    <wsdl:input message="tns:LsRequestSecurityTokenSoapIn" />
    <wsdl:output message="tns:LsRequestSecurityTokenSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="RequestSecurityTokenWithToken">
    <wsdl:input message="tns:RequestSecurityTokenWithTokenSoapIn" />
    <wsdl:output message="tns:RequestSecurityTokenWithTokenSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="LsRequestSecurityTokenWithCookie">
    <wsdl:input message="tns:LsRequestSecurityTokenWithCookieSoapIn" />
    <wsdl:output message="tns:LsRequestSecurityTokenWithCookieSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetProxyTrustConfiguration">
    <wsdl:input message="tns:GetProxyTrustConfigurationSoapIn" />
    <wsdl:output message="tns:GetProxyTrustConfigurationSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetFsTrustInformation">
    <wsdl:input message="tns:GetFsTrustInformationSoapIn" />
    <wsdl:output message="tns:GetFsTrustInformationSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetTrustedRealmUri">
    <wsdl:input message="tns:GetTrustedRealmUriSoapIn" />
    <wsdl:output message="tns:GetTrustedRealmUriSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetClaims">
    <wsdl:input message="tns:GetClaimsSoapIn" />
    <wsdl:output message="tns:GetClaimsSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="FederationServerServiceSoap" type="tns:FederationServerServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="LsRequestSecurityToken">
    <soap:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/LsRequestSecurityToken" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="RequestSecurityTokenWithToken">
    <soap:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/RequestSecurityTokenWithToken" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="LsRequestSecurityTokenWithCookie">
    <soap:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/LsRequestSecurityTokenWithCookie" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetProxyTrustConfiguration">
    <soap:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/GetProxyTrustConfiguration" style="document" />

```

```

    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetFsTrustInformation">
    <soap:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/GetFsTrust
      Information" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetTrustedRealmUri">
    <soap:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/GetTrusted
      RealmUri" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetClaims">
    <soap:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/GetClaims"
      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="FederationServerServiceSoap12" type="tns:FederationServerServiceSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="LsRequestSecurityToken">
    <soap12:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/LsRequestS
      ecurityToken" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="RequestSecurityTokenWithToken">
    <soap12:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/RequestSec
      urityTokenWithToken" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="LsRequestSecurityTokenWithCookie">
    <soap12:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/LsRequestS
      ecurityTokenWithCookie" style="document" />

```

```

    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetProxyTrustConfiguration">
    <soap12:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/GetProxyTrustConfiguration" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetFsTrustInformation">
    <soap12:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/GetFsTrustInformation" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetTrustedRealmUri">
    <soap12:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/GetTrustedRealmUri" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetClaims">
    <soap12:operation
      soapAction="http://schemas.microsoft.com/ActiveDirectory/FederationService/2005/07/GetClaims" 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="FederationServerService">
  <wsdl:port name="FederationServerServiceSoap" binding="tns:FederationServerServiceSoap">
    <soap:address location="https://localhost/adfs/fs/federationserversevice.asmx" />
  </wsdl:port>
  <wsdl:port name="FederationServerServiceSoap12"
binding="tns:FederationServerServiceSoap12">
    <soap12:address location="https://localhost/adfs/fs/federationserversevice.asmx" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

7 (Updated Section) Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Windows Server 2003 R2 operating system
- Windows Server 2008 operating system
- Windows Server 2008 R2 operating system
- Windows Server 2012 operating system
- Windows Server 2012 R2 operating system
- Windows Server 2016 operating system
- Windows Server operating system
- Windows Server 2019 operating system
- Windows Server 2022 operating system

▪ Windows Server 2025 operating system

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

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 3.2.3.1: After the data described in section 3.1.1.1 is obtained for the first time via a GetFsTrustInformation exchange, Windows maintains a cached copy of the data described in section 3.1.1.1.

<2> Section 3.2.4.1: Windows emits a GetFsTrustInformation request when the client service is started, and caches the data from the response. If for some reason there is not a cached version of the data described in section 3.1.1.1 available when a security token is received, Windows will attempt to obtain the data again by emitting a GetFsTrustInformation request upon receipt of the security token.

8 Change Tracking

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

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

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

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

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

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

Section	Description	Revision class
7 Appendix B: Product Behavior	Added Windows Server 2025 to the list of applicable products.	Major

9 Index

A

- Abstract data model
 - client 23
 - server 13
 - GetClaims method 14
 - GetFsTrustInformation method 13
 - GetTrustedRealmUri method 14
 - overview 13
- Applicability 10
- Attribute groups 12
- Attributes 12

C

- Capability negotiation 10
- Change tracking 41
- Client
 - abstract data model 23
 - initialization 23
 - GetClaims 23
 - GetFsTrustInformation 23
 - GetTrustedRealmUri 23
 - overview 15
 - local events (section 3.1.6 22, section 3.2.6 26)
 - message processing 23
 - GetClaims 25
 - GetFsTrustInformation 23
 - GetTrustedRealmUri 25
 - overview 23
 - sequencing rules 23
 - GetClaims 25
 - GetFsTrustInformation 23
 - GetTrustedRealmUri 25
 - overview 23
 - timer events (section 3.1.5 22, section 3.2.5 26)
 - timers (section 3.1.2 14, section 3.2.2 23)
- Complex types 11

D

- Data model - abstract
 - client 23
 - server 13
 - GetClaims method 14
 - GetFsTrustInformation method 13
 - GetTrustedRealmUri method 14
 - overview 13

E

- Events
 - local
 - client 22
 - server 26
 - local - client 26
 - local - server 22
 - timer
 - client 22
 - server 26
 - timer - client 26
 - timer - server 22

Examples

- GetClaims
 - request message 28
 - response message 29
- GetFsTrustInformation
 - request message 27
 - response message 27
- GetTrustedRealmUri
 - request message 28
 - response message 28

F

- Fields - vendor-extensible 10
- Full WSDL 31

G

- GetClaims
 - request message example 28
 - response message example 29
- GetFsTrustInformation
 - request message example 27
 - response message example 27
- GetTrustedRealmUri
 - request message example 28
 - response message example 28
- Glossary 6
- Groups 12

I

- Implementer - security considerations 30
- Index of security parameters 30
- Informative references 8
- Initialization
 - client 23
 - GetClaims 23
 - GetFsTrustInformation 23
 - GetTrustedRealmUri 23
 - overview 15
 - server (section 3.1.3 15, section 3.2.3 23)
- Introduction 6

L

- Local events
 - client (section 3.1.6 22, section 3.2.6 26)
 - server (section 3.1.6 22, section 3.2.6 26)

M

- Message processing
 - client 23
 - GetClaims 25
 - GetFsTrustInformation 23
 - GetTrustedRealmUri 25
 - overview 23
 - server 15
 - GetClaims exchange 21
 - GetFsTrustInformation exchange 15
 - GetTrustedRealmUri exchange 19
 - overview 15
- Messages

- attribute groups 12
- attributes 12
- complex types 11
- elements 11
- enumerated 11
- groups 12
- namespaces 11
- simple types 12
- syntax 11
- transport 11

N

- Namespaces 11
- Normative references 7

O

- Operations
 - GetClaims 21
 - GetFsTrustInformation 15
 - GetTrustedRealmUri 19
- Overview (synopsis) 8

P

- Parameters - security index 30
- Preconditions 9
- Prerequisites 9
- Product behavior 40
- Protocol Details
 - overview 13

R

- References 7
 - informative 8
 - normative 7
- Relationship to other protocols 9

S

- Security
 - implementer considerations 30
 - parameter index 30
- Sequencing rules
 - client 23
 - GetClaims 25
 - GetFsTrustInformation 23
 - GetTrustedRealmUri 25
 - overview 23
 - server 15
 - GetClaims exchange 21
 - GetFsTrustInformation exchange 15
 - GetTrustedRealmUri exchange 19
 - overview 15
- Server
 - abstract data model 13
 - GetClaims method 14
 - GetFsTrustInformation method 13
 - GetTrustedRealmUri method 14
 - overview 13
 - GetClaims operation 21
 - GetFsTrustInformation operation 15

- GetTrustedRealmUri operation 19
- initialization (section 3.1.3 15, section 3.2.3 23)
- local events (section 3.1.6 22, section 3.2.6 26)
- message processing 15
 - GetClaims exchange 21
 - GetFsTrustInformation exchange 15
 - GetTrustedRealmUri exchange 19
 - overview 15
- overview 13
- sequencing rules 15
 - GetClaims exchange 21
 - GetFsTrustInformation exchange 15
 - GetTrustedRealmUri exchange 19
 - overview 15
- timer events (section 3.1.5 22, section 3.2.5 26)
- timers (section 3.1.2 14, section 3.2.2 23)
- Simple types 12
- Standards assignments 10
- Syntax
 - messages - overview 11
- Syntax - messages - overview 11

T

- Timer events
 - client (section 3.1.5 22, section 3.2.5 26)
 - server (section 3.1.5 22, section 3.2.5 26)
- Timers
 - client (section 3.1.2 14, section 3.2.2 23)
 - server (section 3.1.2 14, section 3.2.2 23)
- Tracking changes 41
- Transport 11
- Types
 - complex 11
 - simple 12

V

- Vendor-extensible fields 10
- Versioning 10

W

- WSDL 31