

## [MS-SAMLPR-Diff]:

# Security Assertion Markup Language (SAML) Proxy Request Signing 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 as well as overviews of the interaction among each of these technologies~~ 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 ~~may~~ can make copies of it in order to develop implementations of the technologies ~~that are~~ described in ~~the Open Specifications~~ this documentation and ~~may~~ can distribute portions of it in your implementations ~~using that use~~ these technologies or ~~in~~ your documentation as necessary to properly document the implementation. You ~~may~~ can also distribute in your implementation, with or without modification, any ~~schema, IDL's~~ 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 ~~may~~ might cover your implementations of the technologies described in the Open Specifications ~~documentation.~~ Neither this notice nor Microsoft's delivery of ~~the~~ this documentation grants any licenses under those ~~patents~~ or any other Microsoft patents. However, a given Open ~~Specification~~ Specification ~~may~~ 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 ~~the Open Specifications~~ this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation ~~may~~ might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, ~~e-mail~~ 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~~ standards specifications and network programming art, ~~and~~ assumes, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

Date	Revision History	Revision Class	Comments
3/12/2010	1.0	Major	First Release.
4/23/2010	1.0.1	Editorial	Changed language and formatting in the technical content.
6/4/2010	1.0.2	Editorial	Changed language and formatting in the technical content.
7/16/2010	1.0.2	None	No changes to the meaning, language, or formatting of the technical content.
8/27/2010	1.0.2	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2010	1.0.2	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	1.0.2	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	1.0.2	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	1.0.2	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	1.0.2	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	2.0	Major	Updated and revised the technical content.
6/17/2011	3.0	Major	Updated and revised the technical content.
9/23/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
3/30/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	3.1	Minor	Clarified the meaning of the technical content.
10/25/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	3.1	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	3.1	None	No changes to the meaning, language, or formatting of the technical content.
11/14/2013	3.1	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	3.1	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	3.1	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
6/30/2015	3.1	<del>No</del> <del>Change</del> None	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>7</b>
1.1	Glossary .....	7
1.2	References .....	8
1.2.1	Normative References .....	8
1.2.2	Informative References .....	9
1.3	Overview .....	9
1.4	Relationship to Other Protocols .....	10
1.5	Prerequisites/Preconditions .....	10
1.6	Applicability Statement .....	10
1.7	Versioning and Capability Negotiation .....	10
1.8	Vendor-Extensible Fields .....	11
1.9	Standards Assignments.....	11
<b>2</b>	<b>Messages.....</b>	<b>12</b>
2.1	Transport .....	12
2.2	Common Message Syntax .....	12
2.2.1	Namespaces .....	12
2.2.2	Messages.....	12
2.2.2.1	SignMessageRequest .....	13
2.2.2.2	SignMessageResponse .....	14
2.2.2.3	VerifyMessageRequest .....	14
2.2.2.4	VerifyMessageResponse .....	15
2.2.2.5	IssueRequest.....	15
2.2.2.6	IssueResponse.....	16
2.2.2.7	LogoutRequest.....	16
2.2.2.8	LogoutResponse .....	17
2.2.2.9	CreateErrorMessageRequest .....	17
2.2.2.10	CreateErrorMessageResponse .....	18
2.2.3	Elements .....	19
2.2.4	Complex Types.....	19
2.2.4.1	RequestType .....	19
2.2.4.2	ResponseType .....	19
2.2.4.3	PrincipalType.....	19
2.2.4.4	SamlMessageType.....	20
2.2.4.5	PostBindingType .....	20
2.2.4.6	RedirectBindingType.....	20
2.2.5	Simple Types .....	21
2.2.5.1	LogoutStatusType .....	21
2.2.5.2	PrincipalTypes .....	22
2.2.6	Attributes .....	22
2.2.7	Groups .....	22
2.2.8	Attribute Groups.....	22
<b>3</b>	<b>Protocol Details .....</b>	<b>23</b>
3.1	Common Details .....	23
3.1.1	Abstract Data Model.....	23
3.1.2	Timers .....	23
3.1.3	Initialization.....	23
3.1.4	Message Processing Events and Sequencing Rules .....	23
3.1.4.1	SignMessage .....	24
3.1.4.1.1	Messages .....	24
3.1.4.1.1.1	SignMessageRequest.....	24
3.1.4.1.1.2	SignMessageResponse.....	24
3.1.4.2	VerifyMessage .....	24
3.1.4.2.1	Messages .....	24

3.1.4.2.1.1	VerifyMessageRequest .....	24
3.1.4.2.1.2	VerifyMessageResponse .....	24
3.1.4.3	Issue .....	25
3.1.4.3.1	Messages .....	25
3.1.4.3.1.1	IssueRequest .....	25
3.1.4.3.1.2	IssueResponse .....	25
3.1.4.4	Logout .....	25
3.1.4.4.1	Messages .....	25
3.1.4.4.1.1	LogoutRequest .....	25
3.1.4.4.1.2	LogoutResponse .....	25
3.1.4.5	CreateErrorMessage .....	25
3.1.4.5.1	Messages .....	26
3.1.4.5.1.1	CreateErrorMessageRequest .....	26
3.1.4.5.1.2	CreateErrorMessageResponse .....	26
3.1.4.6	Types Common to Multiple Operations .....	26
3.1.4.6.1	Complex Types .....	26
3.1.4.6.1.1	PrincipalType .....	26
3.1.4.6.1.2	SamlMessageType .....	26
3.1.4.6.1.3	PostBindingType .....	26
3.1.4.6.1.4	RedirectBindingType .....	27
3.1.4.6.2	Simple Types .....	27
3.1.4.6.2.1	LogoutStatusType .....	27
3.1.4.6.2.2	PrincipalTypes .....	27
3.1.4.7	Status Codes for Operations .....	27
3.1.4.7.1	Element <Status> .....	27
3.1.4.7.2	Element <StatusCode> .....	28
3.1.4.7.3	Element <StatusMessage> .....	30
3.1.4.7.4	Element <StatusDetail> .....	30
3.1.5	Timer Events .....	30
3.1.6	Other Local Events .....	30
3.2	Server Details .....	30
3.2.1	Abstract Data Model .....	30
3.2.2	Timers .....	30
3.2.3	Initialization .....	31
3.2.4	Message Processing Events and Sequencing Rules .....	31
3.2.5	Timer Events .....	31
3.2.6	Other Local Events .....	31
3.3	Client Details .....	31
3.3.1	Abstract Data Model .....	31
3.3.2	Timers .....	31
3.3.3	Initialization .....	31
3.3.4	Message Processing Events and Sequencing Rules .....	31
3.3.5	Timer Events .....	31
3.3.6	Other Local Events .....	32
<b>4</b>	<b>Protocol Examples .....</b>	<b>33</b>
4.1	Issue Operation Examples .....	33
4.1.1	IssueRequest Example .....	33
4.1.2	IssueResponse Example .....	34
4.1.3	IssueResponse Example Using Artifact Binding .....	35
4.2	CreateErrorMessage Operation Examples .....	36
4.2.1	CreateErrorMessageRequest Example .....	36
4.2.2	CreateErrorMessageResponse Example .....	37
4.3	SignMessage Operation Examples .....	37
4.3.1	SignMessageRequest Example .....	37
4.3.2	SignMessageResponse Example .....	38
4.4	VerifyMessage Operation Examples .....	39
4.4.1	VerifyMessageRequest Example .....	39

4.4.2	VerifyMessageResponse Example.....	40
4.4.3	VerifyMessageResponse Example Using Redirect Binding .....	40
4.5	Logout Operations Examples .....	41
4.5.1	LogoutRequest Example .....	41
4.5.2	LogoutResponse Example .....	42
4.5.3	LogoutRequest Example - Locally Initiated .....	43
4.5.4	LogoutResponse Example:Final Response to Locally Initiated Request .....	43
4.5.5	LogoutRequest Example with SAMLResponse and RelayState .....	43
4.5.6	LogoutResponse Example with SAMLRequest and RelayState .....	45
<b>5</b>	<b>Security .....</b>	<b>46</b>
5.1	Security Considerations for Implementers .....	46
5.2	Index of Security Parameters .....	46
<b>6</b>	<b>Appendix A: Full WSDL .....</b>	<b>47</b>
<b>7</b>	<b>Appendix B: Product Behavior .....</b>	<b>48</b>
<b>8</b>	<b>Change Tracking.....</b>	<b>49</b>
<b>9</b>	<b>Index.....</b>	<b>50</b>

# 1 Introduction

This document specifies the Security Assertion Markup Language (SAML) Proxy Request Signing Protocol, which allows proxy servers to perform operations that require knowledge of configured keys and other state information about federated sites known by the **Security Token service** **Service (STS)** server.

Sections [1.5](#), [1.8](#), [1.9](#), [2](#), and [3](#) of this specification are normative ~~and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms.~~ All other sections and examples in this specification are informative.

## 1.1 Glossary

~~The~~This document uses the following terms ~~are specific to this document:~~

**Active Directory Federation Services (AD FS) Proxy Server:** An AD FS 2.0 service that processes SAML Federation Protocol messages. **AD FS proxy servers** are clients for the Security Assertion Markup Language (SAML) Proxy Request Signing Protocol (SAMLPR).

**Active Directory Federation Services (AD FS) Security Token Service (STS) ~~Server:~~** An AD FS 2.0 service that holds configuration information about federated sites. **AD FS STS** servers are servers for the Security Assertion Markup Language (SAML) Proxy Request Signing Protocol (SAMLPR).

**certificate:** A certificate is a collection of attributes (1) and extensions that can be stored persistently. The set of attributes in a certificate can vary depending on the intended usage of the certificate. A certificate securely binds a public key to the entity that holds the corresponding private key. A certificate is commonly used for authentication (2) and secure exchange of information on open networks, such as the Internet, extranets, and intranets. Certificates are digitally signed by the issuing certification authority (CA) and can be issued for a user, a computer, or a service. The most widely accepted format for certificates is defined by the ITU-T X.509 version 3 international standards. For more information about attributes and extensions, see [RFC3280] and [X509] sections 7 and 8.

**SAML Artifact Binding:** A method of transmitting **SAML messages** via references in HTTP messages, as specified in [SamlBinding] section 3.6.

**SAML Identity Provider (IdP):** A provider of **SAML** assertions, as specified in [SAMLCore2] section 2.

**SAML Message:** A **SAML** protocol message, as specified in [SAMLCore2] and [SamlBinding].

**SAML Post Binding:** A method of transmitting **SAML messages** via HTTP POST actions, as specified in [SamlBinding] section 3.5.

**SAML Redirect Binding:** A method of transmitting **SAML messages** via HTTP redirects, as specified in [SamlBinding] section 3.4.

**SAML Service Provider (SP):** A consumer of **SAML** assertions, as specified in [SAMLCore2] section 2.

**Security Assertion Markup Language (SAML):** The set of specifications that describe security assertions encoded in XML, profiles for attaching assertions to protocols and frameworks, request/response protocols used to obtain assertions, and the protocol bindings to transfer protocols, such as **SOAP** and HTTP.

**security token service (STS):** A web service that issues ~~claims (2) and packages them in encrypted security tokens~~ security tokens. That is, it makes assertions based on evidence that it trusts; these assertions are for consumption by whoever trusts it.

**SHA-1 hash:** A hashing algorithm as specified in [FIPS180-2] that was developed by the National Institute of Standards and Technology (NIST) and the National Security Agency (NSA).

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

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

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

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

**Web 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.

**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: Schema (XSD):** A ~~description of a type of XML document language that is typically expressed in terms of constraints on~~ defines the ~~structure~~ elements, attributes, namespaces, and ~~content of data types for XML documents of that type, in addition to the basic syntax constraints that are imposed~~ as defined by XML itself. [XMLSCHEMA1/2] and [W3C-XSD] standards. An XML schema ~~provides a view of a document type at a relatively high level of abstraction~~ 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 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 dohelp@microsoft.com. We will assist you in finding the relevant information.



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

[SamlBinding] Cantor, S., Hirsch, F., Kemp, J., et al., "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0", March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf>

[SAMLCore2] Cantor, S., Kemp, J., Philpott, R., and Maler, E., Eds., "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0", March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>

[SOAP1.2-1/2003] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624>

[WSAddressing] Box, D., et al., "Web Services Addressing (WS-Addressing)", August 2004, <http://www.w3.org/Submission/ws-addressing/>

[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>

[WSSC1.3] Lawrence, K., Kaler, C., Nadalin, A., et al., "WS-SecureConversation 1.3", March 2007, <http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.html>

[WSSU1.0] OASIS Standard, "WS Security Utility 1.0", 2004, <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd>

[WSTrust] IBM, Microsoft, Nortel, VeriSign, "WS-Trust V1.0", February 2005, <http://specs.xmlsoap.org/ws/2005/02/trust/WS-Trust.pdf>

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

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

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

## 1.2.2 Informative References

~~None.~~

[\[WS-Trust1.3\] Nadalin, A., Goodner, M., Gudgin, M., Barbir, A., Granqvist, H., "WS-Trust 1.3", OASIS Standard 19 March 2007, http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html](http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html)

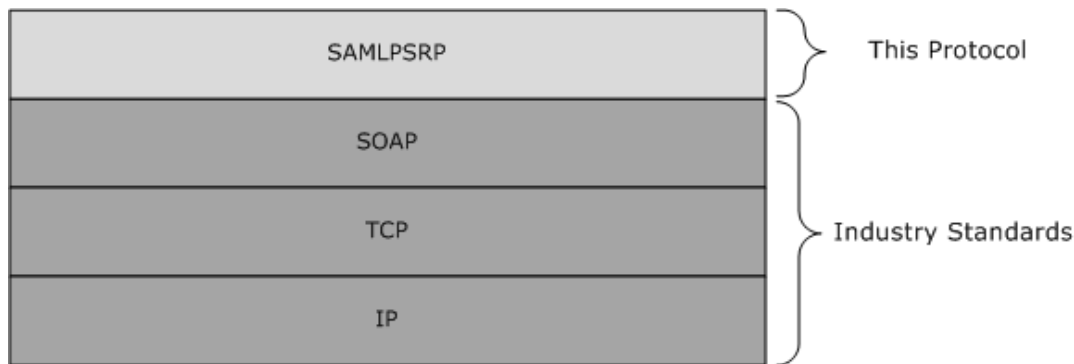
## 1.3 Overview

The Security Assertion Markup Language (SAML) Proxy Request Signing Protocol (SAMLPR) provides the capability for **AD FS proxy servers** to have the **AD FS STS** server for an installation perform operations that require knowledge of the configured keys and other state information about federated sites known by the Security Token Service (STS) server. [For more information, see \[WS-Trust1.3\].](#) In particular, proxy servers use the SAMLPR Protocol to have the STS server in an installation perform **SAML** (see [SAMLCore2] and [SamlBinding]) signature operations upon messages to be sent. Multiple proxy servers [maycan](#) use a single STS server.

The protocol is stateless, with the parameters of each message being fully self-contained.

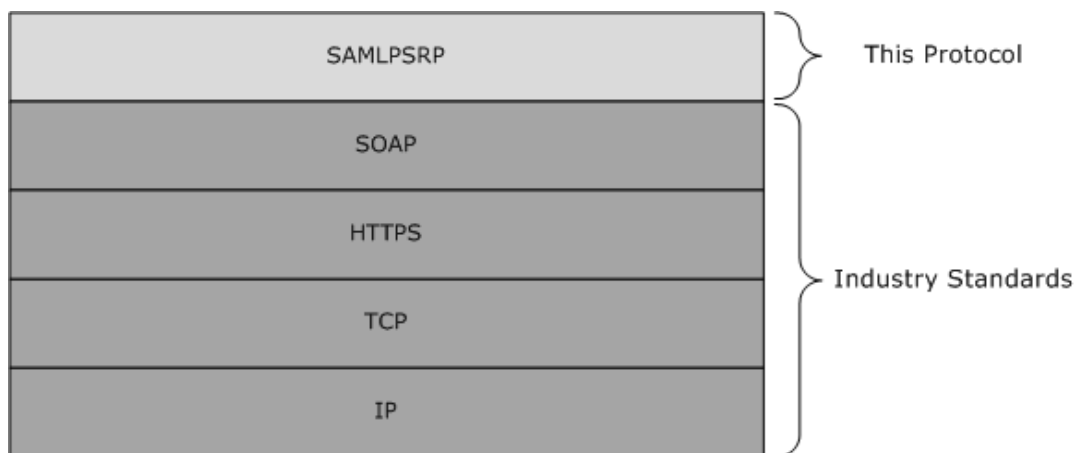
## 1.4 Relationship to Other Protocols

The Security Assertion Markup Language (SAML) Proxy Request Signing Protocol (SAMLPR) uses **SOAP** over TCP for local connections, as shown in the following layering diagram:



**Figure 1: SAMLPR SOAP over TCP layer diagram**

The Security Assertion Markup Language (SAML) Proxy Request Signing Protocol (SAMLPR) uses SOAP over HTTPS for remote connections, as shown in the following layering diagram:



**Figure 2: SAMLPR SOAP over HTTPS layer diagram**

## 1.5 Prerequisites/Preconditions

The client is configured with the **Uniform Resource Locator (URL)** of the server's SOAP service in order to call the service.

## 1.6 Applicability Statement

The SAMLPR Protocol is used by services that perform SAML signature operations for proxy servers by STS servers in a manner that is compatible with AD FS 2.0.

## 1.7 Versioning and Capability Negotiation

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

- SOAP 1.2, as specified in [SOAP1.2-1/2003].

This protocol does not perform any capability negotiation.

## **1.8 Vendor-Extensible Fields**

The schema for this protocol provides for extensibility points for additional elements to be added to each **SOAP message** body. Elements within these extensibility points that are not understood are ignored.

## **1.9 Standards Assignments**

There are no standards assignments for this protocol beyond those defined in the following specification:

- SOAP 1.2, as specified in [SOAP1.2-1/2003].

## 2 Messages

### 2.1 Transport

The Security Assertion Markup Language (SAML) Proxy Request Signing Protocol uses SOAP, as specified in [SOAP1.2-1/2003], over TCP locally or HTTPS remotely, for communication.

### 2.2 Common Message Syntax

This section contains no common definitions used by this protocol.

#### 2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
s	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>	[SOAP1.2-1/2003]
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	[XMLSCHEMA1] and [XMLSCHEMA2]
a	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a>	[WSAddressing] section 1.2
msis	<a href="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol</a>	This document ([MS-SAMLPR])
samlp	<a href="urn:oasis:names:tc:SAML:2.0:protocol">urn:oasis:names:tc:SAML:2.0:protocol</a>	[SAMLCore2]
saml	<a href="urn:oasis:names:tc:SAML:2.0:assertion">urn:oasis:names:tc:SAML:2.0:assertion</a>	[SAMLCore2]
wst	<a href="http://docs.oasis-open.org/ws-sx/ws-trust/200512">http://docs.oasis-open.org/ws-sx/ws-trust/200512</a>	[WSTrust]
wssc	<a href="http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512">http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512</a>	[WSSC1.3]
wssu	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd</a>	[WSSU1.0]

#### 2.2.2 Messages

Message	Description
SignMessageRequest	A message that requests that a <b>SAML Message</b> signature be applied to a SAML Message, if the configuration for the requested principal specifies that messages are to be signed.
SignMessageResponse	A reply message to SignMessageRequest, containing the resulting SAML Message, which is signed, if the configuration for the requested principal specifies that messages are to be signed.
VerifyMessageRequest	A message that requests verification that a SAML Message is from a known party and signed according to the metadata directives for that party.

Message	Description
VerifyMessageResponse	A reply message to the VerifyMessageRequest message, containing a Boolean result.
IssueRequest	A message requesting issuance of a SAML token.
IssueResponse	A reply message to the IssueRequest message containing a SAML response message.
LogoutRequest	A message requesting that a SAML logout be performed.
LogoutResponse	A reply message to the LogoutRequest message containing updated SessionState and LogoutState values.
CreateErrorMessageRequest	A message that requests creation of a SAML error message, which will be signed, if the configuration for the requested principal specifies that messages are to be signed.
CreateErrorMessageResponse	A reply message to the CreateErrorMessageRequest message containing the created SAML error message.

### 2.2.2.1 SignMessageRequest

The SignMessageRequest message requests that a SAML Message signature be applied to a SAML Message, if the configuration for the requested principal specifies that messages are to be signed. It is used by the following message:

Message type	Action URI
Request	http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequest

**body:** The **SOAP body** MUST contain a single msis:SignMessageRequest element with the following type:

```

<complexType name="SignMessageRequestType">
  <complexContent>
    <extension base="msis:RequestType">
      <sequence>
        <element name="ActivityId" type="string"/>
        <element name="Message" type="msis:SamlMessageType"/>
        <element name="Principal" type="msis:PrincipalType"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

**ActivityId:** An opaque string supplied by the caller to track the activity to which this message pertains.

**Message:** A complex type representing a SAML Protocol message.

**Principal:** A complex type representing a SAML EntityId for a **SAML Identity Provider (IdP)**, a **SAML Service Provider (SP)**, or this STS server.

### 2.2.2.2 SignMessageResponse

A SignMessageResponse message is a reply message to SignMessageRequest, containing the resulting SAML Message, which is signed, if the configuration for the requested principal specifies that messages are to be signed. It is used by the following message:

Message type	Action URI
Response	http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequestResponse

**body:** The SOAP body MUST contain a single msis:SignMessageResponse element with the following type:

```
<complexType name="SignMessageResponseType">
  <complexContent>
    <extension base="msis:ResponseType">
      <sequence>
        <element name="Message" type="msis:SamlMessageType"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"
      />
    </sequence>
  </extension>
</complexContent>
</complexType>
```

**Message:** A complex type representing a SAML Protocol message.

### 2.2.2.3 VerifyMessageRequest

The VerifyMessageRequest message requests verification that a SAML Message is from a known party and signed according to the metadata directives for that party. It is used by the following message:

Message type	Action URI
Request	http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequest

**body:** The SOAP body MUST contain a single msis:VerifyMessageRequest element with the following type:

```
<complexType name="VerifyMessageRequestType" >
  <complexContent>
    <extension base="msis:RequestType">
      <sequence>
        <element name="ActivityId" type="string"/>
        <element name="Message" type="msis:SamlMessageType"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"
      />
    </sequence>
  </extension>
</complexContent>
</complexType>
```

**ActivityId:** An opaque string supplied by the caller to track the activity to which this message pertains.

**Message:** A complex type representing a SAML Protocol message.

### 2.2.2.4 VerifyMessageResponse

The VerifyMessageResponse message is a reply to VerifyMessageRequest, containing a Boolean result. It is used by the following message:

Message type	Action URI
Response	http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequestResponse

**body:** The SOAP body MUST contain a single `msis:VerifyMessageResponse` element with the following type:

```
<complexType name="VerifyMessageResponseType" >
  <complexContent>
    <extension base="msis:ResponseType">
      <sequence>
        <element name="IsVerified" type="boolean"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"
      />
    </sequence>
  </extension>
</complexContent>
</complexType>
```

**IsVerified:** A Boolean result indicating whether a SAML Message is from a known party and signed according to the metadata directives for that party.

### 2.2.2.5 IssueRequest

The IssueRequest message requests the issuance of a SAML token. It is used by the following message:

Message type	Action URI
Request	http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequest

**body:** The SOAP body MUST contain a single `msis:IssueRequest` element with the following type:

```
<complexType name="IssueRequestType" >
  <complexContent>
    <extension base="msis:RequestType">
      <sequence>
        <element name="ActivityId" type="string"/>
        <element name="Message" type="msis:SamlMessageType"/>
        <element name="OnBehalfOf" type="wst:OnBehalfOfType"/>
        <element name="SessionState" type="string"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"
      />
    </sequence>
  </extension>
</complexContent>
</complexType>
```

**ActivityId:** An opaque string supplied by the caller to track the activity to which this message pertains.

**Message:** A complex type representing a SAML Protocol message.

**OnBehalfOf:** A complex type representing the party to issue the token for.

**SessionState:** A structured string representing the information required to log out from this session.

### 2.2.2.6 IssueResponse

The IssueResponse message is a reply to IssueRequest, containing a SAML response message. It is used by the following message:

Message type	Action URI
Response	http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequestResponse

**body:** The SOAP body MUST contain a single msis:IssueResponse element with the following type:

```
<complexType name="IssueResponseType">
  <complexContent>
    <extension base="msis:ResponseType">
      <sequence>
        <element name="Message" minOccurs="0" type="msis:SamlMessageType"/>
        <element name="SessionState" type="string"/>
        <element name="AuthenticatingProvider" type="string"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"
      />
    </sequence>
  </extension>
</complexContent>
</complexType>
```

**Message:** A complex type representing a SAML Protocol message.

**SessionState:** A structured string representing the information required to log out from this session.

**AuthenticatingProvider:** The URI of a claims provider or a local STS identifier, depending upon where the user authenticated.

### 2.2.2.7 LogoutRequest

The LogoutRequest message requests that a SAML logout be performed. It is used by the following message:

Message type	Action URI
Request	http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequest

**body:** The SOAP body MUST contain a single msis:LogoutRequest element with the following type:

```
<complexType name="LogoutRequestType" >
  <complexContent>
    <extension base="msis:RequestType">
      <sequence>
        <element name="ActivityId" type="string"/>
        <element name="Message" minOccurs="0" type="msis:SamlMessageType"/>
        <element name="SessionState" type="string"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```



```

        <element name="LogoutState" type="string"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"
/>
    </sequence>
</extension>
</complexContent>
</complexType>

```

**ActivityId:** An opaque string supplied by the caller to track the activity that this message pertains to.

**Message:** A complex type representing a SAML protocol message.

**SessionState:** A structured string representing the information required to log out from this session.

**LogoutState:** A structured string representing additional information required to log out from this session.

### 2.2.2.8 LogoutResponse

The LogoutResponse message is a reply to LogoutRequest, containing updated SessionState and LogoutState values. It is used by the following message:

Message type	Action URI
Response	http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequestResponse

**body:** The SOAP body MUST contain a single msis:LogoutResponse element with the following type:

```

<complexType name="LogoutResponseType">
  <complexContent>
    <extension base="msis:ResponseType">
      <sequence>
        <element name="LogoutStatus" type="msis:LogoutStatusType"/>
        <element name="Message" type="msis:SamlMessageType" minOccurs="0"/>
        <element name="SessionState" type="string"/>
        <element name="LogoutState" type="string"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"
/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

**LogoutStatus:** A complex type representing the status of the logout process.

**Message:** A complex type representing a SAML Protocol message.

**SessionState:** A structured string representing the information required to log out from this session.

**LogoutState:** A structured string representing additional information required to log out from this session.

### 2.2.2.9 CreateErrorMessageRequest

The CreateErrorMessageRequest message requests the creation of a SAML error message, which will be signed, if the configuration for the requested principal specifies that messages are to be signed. It is used by the following message:

Message type	Action URI
Request	http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequest

**body:** The SOAP body MUST contain a single `msis:CreateErrorMessageRequest` element with the following type:

```
<complexType name="CreateErrorMessageRequestType">
  <complexContent>
    <extension base="msis:RequestType">
      <sequence>
        <element name="ActivityId" type="string"/>
        <element name="Message" type="msis:SamlMessageType"/>
        <element name="Principal" type="msis:PrincipalType"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"
      />
    </sequence>
  </extension>
</complexContent>
</complexType>
```

**ActivityId:** An opaque string supplied by the caller to track the activity to which this message pertains.

**Message:** A complex type representing a SAML Protocol message.

**Principal:** A complex type representing a SAML EntityId for a SAML IdP, a SAML SP, or this STS server.

### 2.2.2.10 CreateErrorMessageResponse

The `CreateErrorMessageResponse` message is a reply to `CreateErrorMessageRequest`, containing the created SAML error message. It is used by the following messages:

Message type	Action URI
Response	http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequestResponse

**body:** The SOAP body MUST contain a single `msis:CreateErrorMessageResponse` element with the following type:

```
<complexType name="CreateErrorMessageResponseType">
  <complexContent>
    <extension base="msis:ResponseType">
      <sequence>
        <element name="Message" type="msis:SamlMessageType"/>
        <any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"
      />
    </sequence>
  </extension>
</complexContent>
</complexType>
```

**Message:** A complex type representing a SAML Protocol message.

## 2.2.3 Elements

This specification does not define any common XML **schemaSchema** element definitions.

## 2.2.4 Complex Types

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

Complex type	Description
RequestType	An abstract type containing protocol request message parameters.
ResponseType	An abstract type containing protocol response messages parameters.
PrincipalType	A structure containing a PrincipalTypes value and an identifier for the principal.
SamlMessageType	A structure containing a representation of a SAML Protocol message.
PostBindingType	A structure containing SAML binding information for a <b>SAML post binding</b> .
RedirectBindingType	A structure containing SAML binding information for a <b>SAML redirect binding</b> .

### 2.2.4.1 RequestType

This abstract type contains request message parameters for messages using this protocol. The schema for this type MUST be as follows:

```
<complexType name="RequestType" abstract="true"/>
```

### 2.2.4.2 ResponseType

This abstract type contains response message parameters for messages using this protocol. The schema for this type MUST be as follows:

```
<complexType name="ResponseType" abstract="true"/>
```

### 2.2.4.3 PrincipalType

This structure contains a PrincipalTypes value and an identifier for the principal. The schema for this type MUST be as follows:

```
<complexType name="PrincipalType">  
  <sequence>  
    <element name="Type" type="msis:PrincipalTypes"/>  
    <element name="Identifier" type="string"/>  
  </sequence>  
</complexType>
```

**Type:** A PrincipalTypes enumeration value identifying the type of the SAML principal.

**Identifier:** An identifier for the SAML principal. This is a SAML EntityId.

#### 2.2.4.4 SamlMessageType

This structure contains a representation of a SAML Protocol message. The schema for this type MUST be as follows:

```
<complexType name="SamlMessageType">
  <sequence>
    <element name="BaseUri" type="anyURI"/>
    <choice>
      <element name="SAMLart" type="string"/>
      <element name="SAMLRequest" type="string"/>
      <element name="SAMLResponse" type="string"/>
    </choice>
    <choice>
      <element name="PostBindingInformation" type="msis:PostBindingType"/>
      <element name="RedirectBindingInformation" type="msis:RedirectBindingType"/>
    </choice>
  </sequence>
</complexType>
```

**BaseUri:** The URL to post message to.

**SAMLart:** A SAML artifact identifier, base64-encoded as per [SamlBinding] section 3.6.

**SAMLRequest:** A SAML request message, base64-encoded as per [SamlBinding] sections 3.4 and 3.5.

**SAMLResponse:** A SAML response message, base64-encoded as per [SamlBinding] sections 3.4 and 3.5.

**PostBindingInformation:** Information about the SAML Message using the SAML post binding, as per [SamlBinding] section 3.5.

**RedirectBindingInformation:** Information about the SAML Message using the SAML redirect binding, as per [SamlBinding] section 3.4.

#### 2.2.4.5 PostBindingType

This structure contains SAML binding information for a SAML post binding. The schema for this type MUST be as follows:

```
<complexType name="PostBindingType">
  <sequence>
    <element name="RelayState" minOccurs="0" type="string"/>
  </sequence>
</complexType>
```

**RelayState:** An opaque BLOB that, if present in the request, MUST be returned in the response, as per [SamlBinding] section 3.5.3.

#### 2.2.4.6 RedirectBindingType

This structure contains SAML binding information for a SAML redirect binding. The schema for this type MUST be as follows:

```

<complexType name="RedirectBindingType">
  <sequence>
    <element name="RelayState" minOccurs="0" type="string"/>
    <sequence minOccurs="0">
      <element name="Signature" type="string"/>
      <element name="SigAlg" type="string"/>
      <element name="QueryStringHash" minOccurs="0" type="string"/>
    </sequence>
  </sequence>
</complexType>

```

**RelayState:** An opaque BLOB that, if present in the request, MUST be returned in the response, as per [SamlBinding] section 3.4.3.

**Signature:** The message signature (if present), encoded as per [SamlBinding] section 3.4.4.1.

**SigAlg:** The message signature algorithm (if present), as per [SamlBinding] section 3.4.4.1.

**QueryStringHash:** A base64-encoded **SHA-1 hash** of the redirect query string (if present), for integrity purposes, as per [SamlBinding] section 3.6.4.

## 2.2.5 Simple Types

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

Simple type	Description
LogoutStatusType	An enumeration of status values for logout operations.
PrincipalTypes	An enumeration of the types of SAML principals.

### 2.2.5.1 LogoutStatusType

This type enumerates the set of status values for logout operations. The schema for this type MUST be as follows:

```

<simpleType name="LogoutStatusType">
  <restriction base="string">
    <enumeration value="InProgress" />
    <enumeration value="LogoutPartial" />
    <enumeration value="LogoutSuccess" />
  </restriction>
</simpleType>

```

**InProgress:** Indicates that more logout work is required to be performed.

**LogoutPartial:** Indicates that the logout process is complete, but all session participants might not have been logged out.

**LogoutSuccess:** Indicates the logout process is complete, with all session participants logged out.

### 2.2.5.2 PrincipalTypes

This type enumerates the set of types of SAML principals. The schema for this type MUST be as follows:

```
<simpleType name="PrincipalTypes">  
  <restriction base="string">  
    <enumeration value="Self" />  
    <enumeration value="Scope" />  
    <enumeration value="Authority" />  
  </restriction>  
</simpleType>
```

**Self:** Indicates that the principal is this STS server.

**Scope:** Indicates that the principal is a SAML Service Provider, identified by an Entity Identifier, as per [SAMLCore2] section 8.3.6.

**Authority:** Indicates that the principal is a SAML Identity Provider, identified by an Entity Identifier, as per [SAMLCore2] section 8.3.6.

### 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

### 3.1 Common Details

This section describes protocol details that are common among multiple port types.

#### 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 SAMLPR Protocol enables proxy servers to have STS servers perform operations requiring state held at the STS server. Other than standard SOAP request/response protocol state that is not specific to this protocol, no state about the protocol is maintained at either the protocol client or server.

#### 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 **may-beare** used by the underlying transport to transmit and receive SOAP messages. The protocol does not include provisions for time-based retry for sending protocol messages.

#### 3.1.3 Initialization

No protocol-specific initialization is required to use this protocol. Standard SOAP bindings **MUST** be established between the client and server before initiating communication.

For clients running on the local machine, the standard STS server SOAP endpoint address is `net.tcp://localhost/samlprotocol`. For clients running on remote machines connecting to a server, the standard STS server SOAP endpoint address is `https://contoso.com/adfs/services/trust/samlprotocol/proxycertificatetransport`, where `contoso.com` represents the server domain name. Other port addresses **MAY** be used by implementations. <1>

#### 3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
SignMessage	This operation causes a SAML Message signature be applied to the supplied SAML Message when the configuration requires signing, with the resulting message being returned as a result.
VerifyMessage	This operation verifies whether a SAML Message is from a known party and signed according to metadata directives for that party, returning the result as a Boolean.
Issue	This operation causes issuance of a SAML token.
Logout	This operation causes a SAML session to be logged out.
CreateErrorMessage	This operation creates a SAML error message, applying a signature, if the configuration for the requested principal specifies that messages are to be signed.

For each operation there is a request and reply message. In all cases, the sequence of operation is that the client sends the request message to the server, which responds with the corresponding reply message. The server MUST accept the request messages and the client MUST accept the corresponding reply messages, when sent in response to a request message. The behavior of any other uses of these messages is undefined.

### 3.1.4.1 SignMessage

This operation causes a SAML Message signature be applied to the supplied SAML Message when the configuration requires signing, with the resulting message being returned as a result. This operation consists of the client sending a SignMessageRequest message to the server, which replies with a SignMessageResponse message.

#### 3.1.4.1.1 Messages

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

Message	Description
SignMessageRequest	Conveys request parameters for SignMessage operation.
SignMessageResponse	Conveys response parameters for SignMessage operation.

##### 3.1.4.1.1.1 SignMessageRequest

This message conveys request parameters for the SignMessage operation.

##### 3.1.4.1.1.2 SignMessageResponse

This message conveys response parameters for the SignMessage operation.

### 3.1.4.2 VerifyMessage

This operation verifies whether a SAML Message is from a known party and signed according to metadata directives for that party, returning the result as a Boolean. This operation consists of the client sending a VerifyMessageRequest message to the server, which replies with a VerifyMessageResponse message.

#### 3.1.4.2.1 Messages

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

Message	Description
VerifyMessageRequest	Conveys request parameters for the VerifyMessage operation.
VerifyMessageResponse	Conveys response parameters for the VerifyMessage operation.

##### 3.1.4.2.1.1 VerifyMessageRequest

This message conveys request parameters for the VerifyMessage operation.

##### 3.1.4.2.1.2 VerifyMessageResponse



This message conveys response parameters for the VerifyMessage operation.

### 3.1.4.3 Issue

This operation causes the issuance of a SAML token. This operation consists of the client sending an IssueRequest message to the server, which replies with an IssueResponse message.

#### 3.1.4.3.1 Messages

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

Message	Description
IssueRequest	Conveys request parameters for the Issue operation.
IssueResponse	Conveys response parameters for the Issue operation.

##### 3.1.4.3.1.1 IssueRequest

This message conveys request parameters for the Issue operation.

##### 3.1.4.3.1.2 IssueResponse

This message conveys response parameters for the Issue operation.

### 3.1.4.4 Logout

This operation causes a SAML session to be logged out. This operation consists of the client sending a LogoutRequest message to the server, which replies with a LogoutResponse message.

#### 3.1.4.4.1 Messages

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

Message	Description
LogoutRequest	Conveys request parameters for the Logout operation.
LogoutResponse	Conveys response parameters for the Logout operation.

##### 3.1.4.4.1.1 LogoutRequest

This message conveys request parameters for the Logout operation.

##### 3.1.4.4.1.2 LogoutResponse

This message conveys response parameters for Logout operation.

### 3.1.4.5 CreateErrorMessage

This operation creates a SAML error message, applying a signature, if the configuration for the requested principal specifies that messages are to be signed. This operation consists of the client

sending a CreateErrorMessageRequest message to the server, which replies with a CreateErrorMessageResponse message.

### 3.1.4.5.1 Messages

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

Message	Description
CreateErrorMessageRequest	Conveys request parameters for the CreateErrorMessage operation.
CreateErrorMessageResponse	Conveys response parameters for the CreateErrorMessage operation.

#### 3.1.4.5.1.1 CreateErrorMessageRequest

This message conveys request parameters for the CreateErrorMessage operation.

#### 3.1.4.5.1.2 CreateErrorMessageResponse

This message conveys response parameters for the CreateErrorMessage operation.

### 3.1.4.6 Types Common to Multiple Operations

This section describes types that are common to multiple operations.

#### 3.1.4.6.1 Complex Types

The following table summarizes the XML schema complex type definitions that are common to multiple operations, the schemas for which are defined in section 2.2.4.

Complex type	Description
PrincipalType	Identifies participant in a SAML federation, including its role.
SamlMessageType	Representation of a SAML Protocol message and the binding used to send it.
PostBindingType	Information about a SAML post binding, which consists of its RelayState, if present.
RedirectBindingType	Information about a SAML redirect binding, which consists of its RelayState, if present, and signature information, if present.

##### 3.1.4.6.1.1 PrincipalType

This complex type identifies participant in a SAML federation, including its role.

##### 3.1.4.6.1.2 SamlMessageType

This complex type specifies the representation of a SAML Protocol message and the binding used to send it.

##### 3.1.4.6.1.3 PostBindingType

This complex type specifies information about a SAML post binding, which consists of its RelayState, if present.

### 3.1.4.6.1.4 RedirectBindingType

This complex type specifies information about a SAML redirect binding, which consists of its RelayState, if present, and signature information, if present.

### 3.1.4.6.2 Simple Types

The following table summarizes the XML schema simple definitions that are common to multiple operations, the schemas for which are defined in section 2.2.5.

Simple type	Description
LogoutStatusType	Indicates whether logout operation has completed or not, and if completed, whether all session participants were logged out.
PrincipalTypes	Identifies role of participant in SAML federation.

#### 3.1.4.6.2.1 LogoutStatusType

This simple type indicates whether logout operation has completed or not, and if completed, whether all session participants were logged out.

#### 3.1.4.6.2.2 PrincipalTypes

This simple type identifies the role of the participant in a SAML federation.

### 3.1.4.7 Status Codes for Operations

This section describes both the <Status> element and the different status codes as specified in [SAMLCore2], section 3.2.2.

#### 3.1.4.7.1 Element <Status>

The <Status> element contains the following three elements:

Element	Required/Optional	Description
<StatusCode>	Required	This element MUST contain a code that represents the status of a request that has been received by the server.
<StatusMessage>	Optional	This element MAY contain a message that is to be returned to the operator.
<StatusDetail>	Optional	This element MAY contain additional information concerning an error condition.

The following schema fragment defines both the <Status> element and its corresponding **StatusType** complex type:

```
<element name="Status" type="samlp:StatusType"/>
<complexType name="StatusType">
  <sequence>
    <element ref="samlp:StatusCode"/>
    <element ref="samlp:StatusMessage" minOccurs="0"/>
    <element ref="samlp:StatusDetail" minOccurs="0"/>
  </sequence>
```

</complexType>

### 3.1.4.7.2 Element <StatusCode>

The <StatusCode> element contains a code or a set of nested codes that represent the status of the request. Every <StatusCode> element has the following attribute:

Attribute	Required/Optional	Description
Value	Required	The status code value. This value MUST contain a URI reference. The Value attribute of the top-level <StatusCode> element MUST be one of the top-level status codes given in this section. Subordinate <StatusCode> elements MAY use second-level status code values given in this section.

The <StatusCode> element MAY contain subordinate second-level <StatusCode> elements that provide additional information on the error condition.

The permissible top-level status codes are:

Status code	Description
urn:oasis:names:tc:SAML:2.0:status:Success	The request succeeded.
urn:oasis:names:tc:SAML:2.0:status:Requester	The request could not be performed due to an error on the part of the requester.
urn:oasis:names:tc:SAML:2.0:status:Responder	The request could not be performed due to an error on the part of the SAML responder or SAML authority.
urn:oasis:names:tc:SAML:2.0:status:VersionMismatch	The SAML responder could not process the request because the version of the request message was incorrect.

The second-level status codes are:

Status code	Description
urn:oasis:names:tc:SAML:2.0:status:AuthnFailed	The responding provider was unable to successfully authenticate the principal.
urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue	Unexpected or invalid content was encountered within a <saml:Attribute> or <saml:AttributeValue> element.
urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy	The responding provider cannot or will not support the requested name identifier policy.
urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext	The specified authentication context requirements cannot be met by the responder.
urn:oasis:names:tc:SAML:2.0:status:NoAvailableIDP	Used by an intermediary to indicate that none of the supported identity provider <Loc> elements in an <IDPList> can be resolved or that none of the supported identity providers are available.
urn:oasis:names:tc:SAML:2.0:status:NoPassive	Indicates that the responding provider cannot authenticate the principal passively, as has been requested.
urn:oasis:names:tc:SAML:2.0:status:NoSupportedIDP	Used by an intermediary to indicate that none

Status code	Description
	of the identity providers in an <IDPList> are supported by the intermediary.
urn:oasis:names:tc:SAML:2.0:status:PartialLogout	Used by a session authority to indicate to a session participant that it was not able to propagate the logout request to all other session participants.
urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded	Indicates that a responding provider cannot authenticate the principal directly and is not permitted to proxy the request further.
urn:oasis:names:tc:SAML:2.0:status:RequestDenied	The SAML responder or SAML authority is able to process the request but has chosen not to respond. This status code MAY be used when there is concern about the security context of the request message or the sequence of request messages received from a particular requester.
urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported	The SAML responder or SAML authority does not support the request.
urn:oasis:names:tc:SAML:2.0:status:RequestVersionDeprecated	The SAML responder cannot process any requests with the protocol version specified in the request.
urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh	The SAML responder cannot process the request because the protocol version specified in the request message is a major upgrade from the highest protocol version supported by the responder.
urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooLow	The SAML responder cannot process the request because the protocol version specified in the request message is too low.
urn:oasis:names:tc:SAML:2.0:status:ResourceNotRecognized	The resource value provided in the request message is invalid or unrecognized.
urn:oasis:names:tc:SAML:2.0:status:TooManyResponses	The response message would contain more elements than the SAML responder is able to return.
urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile	An entity that has no knowledge of a particular attribute profile has been presented with an attribute drawn from that profile.
urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal	The responding provider does not recognize the principal specified or implied by the request.
urn:oasis:names:tc:SAML:2.0:status:UnsupportedBinding	The SAML responder cannot properly fulfill the request using the protocol binding specified in the request.

The following schema fragment defines the <StatusCode> element and its corresponding **StatusCodeType** complex type:

```
<element name="StatusCode" type="samlp:StatusCodeType"/>
<complexType name="StatusCodeType">
  <sequence>
```

```
        <element ref="sampl:StatusCode" minOccurs="0"/>
    </sequence>
    <attribute name="Value" type="anyURI" use="required"/>
</complexType>
```

### 3.1.4.7.3 Element <StatusMessage>

The <StatusMessage> element specifies a message that MAY be returned to an operator. The following schema fragment defines the <StatusMessage> element:

```
<element name="StatusMessage" type="string"/>
```

### 3.1.4.7.4 Element <StatusDetail>

The <StatusDetail> element MAY be used to specify additional information concerning the status of the request. The additional information consists of zero or more elements from any namespace, with no requirement for a schema to be present or for schema validation of the <StatusDetail> contents.

The following schema fragment defines the <StatusDetail> element and its corresponding **StatusDetailType** complex type:

```
<element name="StatusDetail" type="sampl:StatusDetailType"/>
<complexType name="StatusDetailType">
    <sequence>
        <any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
```

## 3.1.5 Timer Events

This protocol does not require timers beyond those that **may beare** used by the underlying transport to transmit and receive **soapSOAP** messages. 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 and TCP. This protocol relies on these transport mechanisms 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 Server Details

### 3.2.1 Abstract Data Model

This port type utilizes the common abstract data model described in section 3.1.1.

### 3.2.2 Timers

This port type utilizes the common timers design described in section 3.1.2.

### 3.2.3 Initialization

This port type utilizes the common initialization design described in section 3.1.3. In addition, an implementation SHOULD publish a SOAP endpoint at the port `net.tcp://localhost/samlprotocol` to be connected to by local clients. Also, an implementation SHOULD publish a SOAP endpoint at the port `https://contoso.com/adfs/services/trust/samlprotocol/proxycertificatetransport`, where `contoso.com` represents the server domain name, to be connected to by remote clients. Other port addresses MAY be used by implementations.<2>

### 3.2.4 Message Processing Events and Sequencing Rules

This port type utilizes the common message processing events and sequencing rules described in section 3.1.4.

### 3.2.5 Timer Events

This port type utilizes the common timer events design described in section 3.1.5.

### 3.2.6 Other Local Events

This port type utilizes the common other local events design described in section 3.1.6.

## 3.3 Client Details

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

### 3.3.1 Abstract Data Model

This port type utilizes the common abstract data model described in section 3.1.1.

### 3.3.2 Timers

This port type utilizes the common timers design described in section 3.1.2.

### 3.3.3 Initialization

This port type utilizes the common initialization design described in section 3.1.3. In addition, an implementation SHOULD connect to a SOAP endpoint at the port `net.tcp://localhost/samlprotocol` for a local connection to the STS or it SHOULD connect to a SOAP endpoint at the port `https://contoso.com/adfs/services/trust/samlprotocol/proxycertificatetransport`, where `contoso.com` represents the STS domain name for a remote connection. Other port addresses MAY be used by implementations.<3>

### 3.3.4 Message Processing Events and Sequencing Rules

This port type utilizes the common message processing events and sequencing rules described in section 3.1.4.

### 3.3.5 Timer Events

This port type utilizes the common timer events design described in section 3.1.5.

### 3.3.6 Other Local Events

This port type utilizes the common other local events design described in section 3.1.6.



## 4 Protocol Examples

### 4.1 Issue Operation Examples

#### 4.1.1 IssueRequest Example

This is an example of a message requesting issuance of a SAML token.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequest</a:Action>
    <a:MessageID>urn:uuid:cc11441e-1d06-45b5-b0b5-ef73eee87659</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">net.tcp://localhost/samlprotocol</a:To>
  </s:Header>
  <s:Body>
    <msis:IssueRequest
xmlns:msis="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
      <msis:ActivityId>00000000-0000-0000-0000-000000000000</msis:ActivityId>
      <msis:Message>
        <msis:BaseUri>http://localhost</msis:BaseUri>

<msis:SAMLRequest>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGlucyZ0idXRmLTE2Ij8+PHNhbWxwOkFldGhuUmVxdWV
zdCBJRd0iX2QzYWNjZWI3LWVlZjctNDI5Ny1iMTgyLWEOmYxYzQ3NWJjMSIgVmVyc2lvbj0iMi4wIiBjc3N1ZULuc3Rh
bnQ9IjIwMDktMTItMThUMDE6MzE6MDYyNDM0WiIgc29uc2VudD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOmNvb
nNlbnQ6dW5zcGVjaWZpZ2WQIiIHhtbG5zOnNhbnWxwPSJlcm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6CHJvdG9jb2wiPj
xJc3N1ZXIgeG1sbnM9InVybjpvYXNpczpuYW11czp0YzpzTUU1M0jIuMDphc3N1cnRpb24iPmhoDHA6Ly9leHRlcm5hbHJ
wL3Njb3BlPC9Jc3N1ZXI+PC9zYWIscDpBdXRoblJlcXVlc3Q+</msis:SAMLRequest>
      <msis:PostBindingInformation></msis:PostBindingInformation>
    </msis:Message>
    <msis:OnBehalfOf>
      <wssc:SecurityContextToken wssu:Id="_7b5d980c-9309-474e-ace8-23a99bbe261d-
6C82EA4288DB37210E653FCF8E064B57" xmlns:wssc="http://docs.oasis-open.org/ws-sx/ws-
securityconversation/200512" xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd">
        <wssc:Identifier>urn:uuid:24e876b6-1b0e-43e4-95da-7de16ec31f76</wssc:Identifier>
        <wssc:Instance>urn:uuid:a27fafd2-7e20-47c5-a004-3d83bed8e8f4</wssc:Instance>
        <mss:Cookie
xmlns:mss="http://schemas.microsoft.com/ws/2006/05/security">WFUABeuNCOWL9thXJ601uZ9/RNRXopMT
MYRhy/PRX3SAAAABK91yIGLJLwXwgu5vEdh3wsm4zf7cBxsK5WaaM5TqQjGDlJ7qhgnpjNBwz9J7r/8fqJLdscGZvU7E
ifqfkkkoXX0IkDf+fUxXr0oBE/dy4BKGrK1SQ7VqOULAR4Xr39+X8Jp/eeMncIaJuZ01DSB4MwulVpZKhc3grjPfAog
1wBwAAmoPlIv2HElhlpYbFBmaYmYzpqCOa/Ptr08YCN8YeH1FzEm9229H5oEG87TMEjYnuAelBAMGo8BhgBtVtS+o16Pj
XCSeLF3J/vabemgbxIfJnqh4x5xuYldIRO9FJH78syGjOtGFAVi3KRnpIvnrPg3YKRW0sknIH2lDDzjaFGPZw/wlB0Yen
bWFH+sRkfd+jOqhTvk+3++oeYCzWWSiAZhWDMZKA/kqv3Rh05Drr0v6JbzS3H+PjzXL1NeEvD8Nzhx+0tINy+I3PWIHX
C7WFgYeS8T1TpaXBq+zrH5DDEQl4haozU+41lT7pBcY9NdljLedSK/Eo5/FyvsM8g2HKL0jgKbr6jB3XYRfFDjAlTIWZ
jq7lZQqqAnaa9XzpcVKxZ4sHySUo1GK6uFYWpdZYUe5sTmW4bdYVJw6bNS0KEzhekLroclE8c9Ldv61idHQJuvG0qT6xU
pCkD6KeUrV6ZQxRoKs5fyemG8sRw7R+p9tLpbPjqnpj4SbAjXwOQJA0ksZ0KCDn+VBQIq/YIc+fWd0Jv6/S+rZLDl1UXg
MYPmdGcfIMFZEMlCjknZ8IdcVGIXAuK9AFAfJpfgA+vQowgqXop6Abi//pKNrDa+ChNOQIkSFQoz5btiOpd63j5dKu/Y5
CqR+thD7eYsrTflzdhw10xyFAn0beoETCRSGsmCgo2iBvWWTxze1rKwPfn5wB0dnznH5ruPAOSQ9alto8k2dqbyavuPri
3MqehLnsBXR9Lh5j45gs19+InjbJv/Se1Xsbh5BkbTzP9pghkG4ALivz1aRbcQUDXe85Tb1hcJyD1AVs1PudCMHD1N8p
DDPAkgAzCihTiBnEl1jHL7uCeC+UKfEu6Hv8N134yw5vRPERgq8VArBoUBXVSq9/p4Adv/HGtJbLU2hLl/rr9DqOru3h
HlpOR18aLlddHt/n004awqaIconXGILFqlwMRJXP3J8JL9CncPbp/eszX83o1GILPR+dnSRnAjHQbcGwKSM/5VmtHRVie
g1mXQVcJ90gctdmPictX8lUaxruGJezeIzkSwjtXHSg5HPEWg91mtx9Snr04BD9XVeIMhKInBcyzKVlglR+AxopfNCqg
sKyA+eYbGVOCscdmLnRl2gsvYlJZBz2uBfgqCq6BLgIOQYnSKdyEPTfcIn6aupftsh5zmnnd/lvXY6b0TCXQ+iNuLm
bx1Ezh2ynbAL3UNqt4hdIQGEfswqR6TPKlp0dpd4T5yGtEccOpfL2nwbICsRLI1SnP4pBRuULw7cnlx4IzJcU+vpLmGs
GpdtSUpJyxu+8XSAAh13wBxv8g+X3sZKNxKDAUncwHiq7QHZPaRRat2S9i87+GJg6CFrfIbh32exctEY4c5eR/yXi8y2s
RTLqmfx2s3+108sDMwCpunHh/ygRwK9NWq8BvuAcPmK5norSia+9//wyeeei9e3Ez3i/iMAWAYVoVYt1uom5jkwHEDR
fLZ0t5lRtejC1kPqFBAgDruJ+T402E3quHeGar1li7XR5YO7EQocv07UGVOJ++YgtXb//SrdIFStO+MiOHv5AOIzDlab+
qKSRRhpSWmXK18x4Rja+5qBDE2+gPffj0lp42YC9ZSVxrhu/yHw/zdNaaf106WAiaehYjIrfMiTx6yIXL0f6re9F9FpY
```



lvS2JQemJ5UWQ1SVRiY1lZSXlpVFBKZ0UrNEkralIyT21leWVHemlzY0hZc2s3MG5wRWxGb1Rk2NXZXZyB3BTd28yRnZ  
jNVF0V3dicHN4UnBXS3E40CtjcxPuv0xoS0lZMG92Y2ZjZv1aWfNm2xpK2NRajVESm5GS1pSenpJmZFoSupaRFJ3OXpM  
TmR6eU8zN1J3RmVwrjhESTF6VDdwdfIXSDJKV3ZNQWlnb29rSWH6ZDFXaEFDSHNNES4Q09nWnZENmh1d1BQYm9vSWNLt  
XJYWmpqkXV1AxZG1pb3JVZ0hZa3czY0xkUzF4bTc5Rk9M221JbWRMcmhSRFFZa0VxeWlRc1g5M2FBVBHTanZvREgzMn  
RSNG1Zc2tnYlMvOFJKAGRHMUxUXJ4dXlSsXQ1MmrdBQvRwPa1zRHFxc0NQY1JxYWFXTNybGJrU010T29sL2JvbU1  
oenRFZi81dwl6UTFvY1AwV2J0UFVneXNPTnhtY29HQ3VIS0xZcDBuRWkxdXhMdfN5R0RyQTJKeGhORnpHb0hraF16Q1Qy  
WWhEM1ZmQ2x5YXR0N1R4OXRvT29qZUJ3bjYyWmxqMWhIdmNFUmFJRVV3VfHmZnoxblm1VldIYURoV1Z1V2Nack140XdIQ  
TNZMGt2L3RJTl1qldrL1dUTEM2d1RRcE16NkxZT1cyazNPdUZJTYtmRWRCYTLXTNNb05HSENOwEs1bEsxTDBmTGdYRG  
NryWxiTXNtNzRhaHE3L2xwZmJyU1F1cGdZY3ZCUJSTmVBeFhGam15Z0F3Mv1RUHNBtm9FWkNJT2VEctJ0a1RqS010eXh  
kdeZTNitKbTBZ2JPb3FueGc3eJJKc1dwYVfocmtDcHgl1LzFXbn11ZVRaMwJmV3cyaxZ4N3hnUjMxZXJ1NUFTTG1wdUtH  
U21LenVFSFA4NHfXr3N1OHA4Y0ExQmhoNzVXMVFBMG9Sc0o1Tzd5OVc3VXQ2dWY5R1hjcnhxcj1LLNzR1WnhJUDBvTEVsZ  
lgzMXo5eW9pbw14WHNBZGtWR0p4bD1jC0ROeTdOcE1iOXBoZmpKTFGTFSEFhYjhQcjI4U0h1NUU0L2ZrcVNXSk9kt2tJcX  
JjUFZ4TV1Lb0pvaE9YU84YmlINnhvdGZHODZRE1lcWo4ZS5TkvYNjv6UHg4cTBxVHY1b1JvQUFFV1YyUUVSc3daU2d  
uTk1OSkRvVwX2MGd41RkNxaWduN05Kc0Q3WVYyWVZHDhntOVJmV2hgBhdON0czMwD1Vk05R1Rfd0PzHa3BLZmU5tVHdm  
WTZjVfJ5eXlZT1ovQ3hHZ2IwYmdkL3hNzm42MG14Y1V1dEtYVUhhkbjRvaG9nUXViUUPhM21TT1FqS1JORG1EOEt0Rmx3M  
GFwD3pCYXyWt0tkU3NCQjAZUGN0YjVGWVZuNm5iNXVpTtNaTW1YbFpOWsVaVdRbnVQT1Vya2hZaVpLNFR3SDBVU4rc3  
4V5YmNKNyt2dmlwenh1MkxLZjF3YisZajNwCXJTZn1BYjVzUW430FpUSF1OUFPfYXU4RH1Ya0E0cEY3MHI2SnVRL2tGS31  
WTVUUYTA4NntbHRQekhaN2Zku2dGVFRTE5IT09BeT13SnJ1VWFEB2RBdmYramY5TV43M31BajZmVrZRXBETZnUdjJ4  
dXRiNHpCNktxSvdkbVNZK3ZkNEJqWDFLbE8xv0tXOUFRN0FaSkFzSTJQMzZhoW9PYVJwTkpITS9yaGJGK1VZcFMvY3pVW  
Ut5NWVYWC9wb2pxellpR0s1VhG10EXpdjMz2M3Z2Wp5aGkzNm95NEg3K1NwWDDJRYW1PNGVYTWRLR21SQ01vd0V6RFTFLQV  
1COE9kY1NRSER4U2JVlyttUGlaNE9QZk5hMvV2RFJCbU43cVd3Tk1LSH1Rc0FDaWmWNE9YWU1CeW0zRE1weWx2SWUva2Z  
RckFkHBRiZa0V3R0xUit1Mk12R1Z4NEFTd1d1WHRSR1NUd1udzRmajhXUf1RdFk1SFzhuUdJwD1dW1dTMuRRL21a  
WmV1dUx4ajfBcFVMZUswK01YcTnHUm1TYTJSyZBYQ2IxY3pLeVpXVE5FVmpORytLL2dxWepSbH1Wm1Zpd0JMK0Z5T09Lc  
3hwT2JkamhHTHNYaTzQR2RWTytMbTRKSOR2eTQreVrwa3FOK3JwTgd5M1VSSFJ3SERUWVNQC2NraG04TVAvbmswT1ZLK2  
Jin2pRZDd4bXbONFVrMXV2aStvcGZSS01FTvo3WjVubFgvSkpOC9DRXk1dTVwMi96ZjVXZjR6eHpmSWExb1p0WkxKVUz  
oMnFoZx1LzNEZy1Ua30V1UblpDdTzLWkU5NEZVHmFjUG1PdTBdVHo4cVM0VUpCL0tqQWV6Q3ZJQ0I2dn1OUndSOWh6  
Ti9vazdVK2NLUHRsK1M3WfNRZDdpN0didFI5S1c2b1ZPQW1yVG05RGM5dTBibzFuNXpPTXpKYUUm1dnOEVGOVJoSTE0c  
nAyQ3NpVdgyQnhFTnRyMXA3L1BtNndCQ1FBbW55RUS0aTddcFNzdHRu0WpZajJGwMnkd0o1REZIRGQzRWZveGRLMkw0Wk  
5kdE9pSUpaeeQ5bhV1QytSNHczT2V3b3pQWEVnd1NjenBirXQyUHNpdnR6VzFFVEYwZU5xcEt1R2F5SEhOUkRML0x6UGP  
KYzd1S1BmUXZ5bHp0TksZTEt1bm9rMkFTUUZyCvHCUgPwWDDNDandOemdOWFdad19xUC9ue11UakpuUdJVFdBmitqNGLR  
VWpCSFAxVfZzK3EvVDUwCtsQ215Qys4Rml1T3Y2R0JNTG1Xd1d3QVFMNHFrV1VuRHU2bnUxSEx1MXZqdXZJcDjwTETYS  
zhmNT1Lem1CdjFHU1VsYnBaY1BkVHJWSHcrcHNyRjNBW1RVU1d0TmPFMHPRY2pMUUVuc1hTRHN6S0dzVmh2d1BOQjhxL2  
VvNnE0dTITzNnBgE5tN0hzRjB3UHfWl0xtQVNZR1RNUDBkQitYMG1FTytLcWgxQzqbkxkWWFVMjJrdCtGcStCbFhhSG1  
MTS0Q1qVnNbhU2VSAU01PWkhjRkFsVEYxUU5iak1SaXdhbnJDQ1JpcHg4b1ZuVEdObWF2RzFVRNqZcnBucGFCRLpnMnNW  
TWpha2kzZkpxUXkxbkVxejBLdHBUWFRaWURjK3I3U1M2TE02K1dXZitpTTR2Qkx1Ly9tejN0SD1haFpob0s1c0NedC9uz  
GJHSUk4emtYbngvai9aUXdtek9vMndwSFJmbXNYs25UMmNRNhp4du51tZVUVVZTmN1cko5dmhidTR30Tc2b2R4K0JJK2  
1QZWTzL0hxTmXibXhYymx2cz1lcTVXWVUmUmhIVjFpd3NITWNKYThnZGdjY1llWN0NsaGRxUitPeVzSVWFtEbEpSemFEYTD  
SWX1TbU1wbzZaT3BUeTdhQWVNaWhnNit4VfHMRVBDZHZCQjzeDUrUmJIRUpQbmJocEpxUFFkc21uWk1jYkZjREp5aFhu  
VmdQVXJGYNi0VVRLbDFYyVdPcDNGUS94c1BHWGFVSmZRN0ZRYLZJNKdoUEFBamVPK1Y1Yw5Wa3NpekpTY0dsNXp50E1GQ  
1E1eEFsQ1pCSFJSLytNMWtvNzNNUnExS8yZzhrR0k1N3gyQWczZlhQsZJvRHVVtkdwVEc5c2JZRMV5VGxCTVBuRmtaTG  
dLT1BhczZLZGEyUmNWZ02Q0tGeCt3bGU1TlQ5TTQ1d1ZGzBBLVEV3UF16QmdTa24zS2NqN0t5NmRUVGxIaUtju0QxaCt  
Pd2FyM2dDZ3lwcGcrMDBDSGL0Z2NOU282cmoxV1Vcmz5STJvQkRQMGJBUC9BS2Mva0MycKriVmRhtGpTbS9FSDducTNW  
OUlzekhhL1YrOVdFmNRVZklqQ1JNRy8venV2WT08L3h1bM6Q21waGvYVmFsdWU+PC94Z5WjOkNpcGh1cRhdGE+PC94Z  
W5jOkVuY3J5cHRlZERhdGE+PC9FbmNyeXB0ZWRBc3N1cnRpb24+PC9zYw1scDpS2XNwb25zZT4=</msis:SAMLRespons  
e>

```
<msis:PostBindingInformation></msis:PostBindingInformation>
</msis:Message>
<msis:SessionState></msis:SessionState>
<msis:AuthenticatingProvider>http://localhost/</msis:AuthenticatingProvider>
</msis:IssueResponse>
</s:Body>
</s:Envelope>
```

### 4.1.3 IssueResponse Example Using Artifact Binding

This is an example of a reply to a request to issue a SAML token, which contains the resulting SAML response message. In this example, the **SAML Artifact Binding** was employed.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/Proc
essRequestResponse</a:Action>
    <a:RelatesTo urn:uuid:0ac7deb2-4d52-4a77-8071-d4bb099e6db9</a:RelatesTo>
    <a:To s:mustUnderstand="1">http://www.w3.org/2005/08/addressing/anonymous</a:To>
```

```

    </s:Header>
    <s:Body>
      <msis:IssueResponse
xmlns:msis="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
        <msis:Message>
          <msis:BaseUri>https://externalrp/</msis:BaseUri>

<msis:SAMLart>AAQAAPbJen9kBjz+58LcIVeEcGTU2/CTgpbO7ZhNzAgEAN1B90ECfpNEVLg=</msis:SAMLart>
          <msis:RedirectBindingInformation></msis:RedirectBindingInformation>
        </msis:Message>
        <msis:SessionState></msis:SessionState>
        <msis:AuthenticatingProvider></msis:AuthenticatingProvider>
      </msis:IssueResponse>
    </s:Body>
  </s:Envelope>

```

## 4.2 CreateErrorMessage Operation Examples

### 4.2.1 CreateErrorMessageRequest Example

This is an example of a message that requests creation of a SAML error message.

```

    <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
      <s:Header>
        <a:Action
s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/Proc
essRequest</a:Action>
        <a:MessageID>urn:uuid:678452fe-e24d-439e-8543-e2e72f936930</a:MessageID>
        <a:ReplyTo>
          <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
        </a:ReplyTo>
        <a:To s:mustUnderstand="1">net.tcp://localhost/samlprotocol</a:To>
      </s:Header>
      <s:Body>
        <msis:CreateErrorMessageRequest
xmlns:msis="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
          <msis:ActivityId>00000000-0000-0000-0000-000000000000</msis:ActivityId>
          <msis:Message>
            <msis:BaseUri>http://localhost</msis:BaseUri>

<msis:SAMLRequest>PD94bWwgdmVyc2lvdj0iMS4wIiBlbmNvZGluZz0idXRmLTE2Ij8+PHNhbWxwOkF1dGhuUmVxdWV
zdCBJRd0iXzIwN2U2YTdhLTA1YTgtNGMzOS1iMTE0LTgyYzc5ZTk1Y2NmOCIGVmVyc2lvdj0iMi4wIiBjc3N1ZUlu
c3RhbnQ9IjIwMDktMTItMTUMDE2MzE6MTEuODYzWiIgc29uc2VudD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOmNvb
nNlbnQ6dW5zcGVjaWZpZWQiIFByb3RvY29sQmluZGluZz0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOmJpbmRpbm
dzOkhUVFAtUmVkaXJlY3QiIHhtbG5zOnNhbWxwPSJ1cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6cHJvdG9jb2wiPjx
Jc3N1ZXIgeG1sbnM9InVybjpvYXNpczpuYW1lc2p0YzptQU1MOjIuMDphc3N1cnRpb24iPmh0dHA6Ly9leHRlcm5hbHJw
L3Njb3B1PC9Jc3N1ZXI+PC9zYW1scDpBdXRoblJlcXVlc3Q+</msis:SAMLRequest>
          <msis:PostBindingInformation></msis:PostBindingInformation>
        </msis:Message>
        <samlp:Status xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
          <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Responder">
            <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext"></samlp:StatusCode>
          </samlp:StatusCode>
        </samlp:Status>
      </msis:CreateErrorMessageRequest>
    </s:Body>
  </s:Envelope>

```

## 4.2.2 CreateErrorMessageResponse Example

This is an example of a reply to a message requesting creation of a SAML error message, which contains the created SAML error message.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequestResponse</a:Action>
    <a:RelatesTo>urn:uuid:678452fe-e24d-439e-8543-e2e72f936930</a:RelatesTo>
    <a:To s:mustUnderstand="1">http://www.w3.org/2005/08/addressing/anonymous</a:To>
  </s:Header>
  <s:Body>
    <msis:CreateErrorMessageResponse
xmlns:msis="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
      <msis:Message>
        <msis:BaseUri>https://externalrp</msis:BaseUri>

<msis:SAMLResponse>7b0HYBxJliUmL23Ke39K9UrX4HShCIBgEyTYkEAQ7MGIzeaS7B1pRyMpqyqBymVWZV1mFkDM7Z
28995777333nvvvfe60510J/ff/z9cZmQBbPbOStrJniGAqsqfP358Hz8iHjfZolw9epU3q2rZ50nZ088++v2zLL+/vzO
5tz3Zy/e29/PJ/elsz7q7ne8/yPbuP9g9mB3MPkp/Mq+bolp+9tHeeOeJ9KxplvnZsmmzZUzf7ew83N7d2949eLOz++je
7qPd3fHDvb2f+ih9mjdtsxafnPetqvm0d27+bs2r5dZWa/ufpSeABEAWdfLR1XWFM2jZbbIm0ft9Nhr4y+eP6L+Hk2l0
aPlslnl0+K8yAmjs6UZx5uKhrG38yD/NHuQbe/czw6296f3Hm5Pdnf3tw/2pg8e5g/vT6fnBx+17xblsnnEZnJc5aqu2m
palR8dPebBlvLq5peypslrDPajIwyWxlpW06ycV0179/FdgXOkk/C6zdp1E/51Us3y9Cezcplv7qfh1jqPs7z+6IPAvKi
01+18STPR0tR81N49eny3C67zkf3TTMHR/wM=</msis:SAMLResponse>
        <msis:RedirectBindingInformation>

<msis:Signature>R1FtupsaiITbNa5wL4+mOnuFpRBYs5kq/ni5ycqNprqp0l0c5+RUOA5/8RkmRY787oB817FfFJOYw
3FkI1hWayPqclb1HFp7AcuJFPmWVT2bGXbdRV6sCFV0g5X0lPsYG+a/9EZdiYUaMCRUvOds0s5SdtmL95FCQpLxkG5PEk
w=</msis:Signature>
        <msis:SigAlg>http://www.w3.org/2001/04/xmldsig-more#rsa-sha256</msis:SigAlg>
      </msis:Message>
    </msis:CreateErrorMessageResponse>
  </s:Body>
</s:Envelope>
```

## 4.3 SignMessage Operation Examples

### 4.3.1 SignMessageRequest Example

This is an example of a message that requests that a SAML Message signature be applied to a SAML Message.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequest</a:Action>
    <a:MessageID>urn:uuid:5654c3f9-691f-4f9e-aa51-d5d37060dc88</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">net.tcp://localhost/samlprotocol</a:To>
  </s:Header>
  <s:Body>
    <msis:SignMessageRequest
xmlns:msis="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
      <msis:ActivityId>00000000-0000-0000-0000-000000000000</msis:ActivityId>
```



```
h2RUovOURvYit0K2l1TUGp1N2VtUkphVEtubnZtaEdQeXRadTVzazlyVDDpMTBIS1NCQkZISTJGZmRBPTwvZHM6WUOWOUN
lcnRpZmljYXRlPjYwZHM6WUOWOURhdGE+PC9LZXlJbWZvPjwvZHM6U2lnbmf0dXJlPjwvc2FtbHA6QXV0aG5SZXF1ZXNO
Pg==</msis:SAMLRequest>
  <msis:PostBindingInformation></msis:PostBindingInformation>
  </msis:Message>
</msis:SignMessageResponse>
</s:Body>
</s:Envelope>
```

## 4.4 VerifyMessage Operation Examples

### 4.4.1 VerifyMessageRequest Example

This is an example of a message that requests verification that a SAML Message is from a known party and signed according to the configuration for that party.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/Proc
essRequest</a:Action>
    <a:MessageID>urn:uuid:05fbb0db-e105-448b-b127-1bf689738d75</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">net.tcp://localhost/samlprotocol</a:To>
  </s:Header>
  <s:Body>
    <msis:VerifyMessageRequest
xmlns:msis="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
      <msis:ActivityId>00000000-0000-0000-0000-000000000000</msis:ActivityId>
      <msis:Message>
        <msis:BaseUri>http://contoso.com/</msis:BaseUri>

<msis:SAMLRequest>PHNhbWxwOkFldGhuUmVxdWVzdCBJRd0iXzA4MTZjZjJiLTg2YzUtNDU2Ny04MGVlLTFkZjVmYjY
jZmYzYiIjYwZHM6WUOWOURhdGE+PC9LZXlJbWZvPjwvZHM6U2lnbmf0dXJlPjwvc2FtbHA6QXV0aG5SZXF1ZXNO
Pg==</msis:SAMLRequest>
      </msis:Message>
    </s:Body>
  </s:Envelope>
```

```

lcnRpZmljYXRlPjwvZHM6WDUwOURhdGE+PC9LZXlJbmZvPjwvZHM6U2lnbmF0dXJlPjwvc2FtbHA6QXV0aG5SZXF1ZXNO
Pg==</msis:SAMLRequest>
  <msis:PostBindingInformation></msis:PostBindingInformation>
</msis:Message>
</msis:VerifyMessageRequest>
</s:Body>
</s:Envelope>

```

#### 4.4.2 VerifyMessageResponse Example

This is an example of a reply to a request to verify that a SAML Message is from a known party and signed according to the configuration for that party, containing the verification response.

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequestResponse</a:Action>
    <a:RelatesTo>urn:uuid:05fbb0db-e105-448b-b127-1bf689738d75</a:RelatesTo>
    <a:To s:mustUnderstand="1">http://www.w3.org/2005/08/addressing/anonymous</a:To>
  </s:Header>
  <s:Body>
    <msis:VerifyMessageResponse
xmlns:s="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
    <msis:IsVerified>true</msis:IsVerified>
    </msis:VerifyMessageResponse>
  </s:Body>
</s:Envelope>

```

#### 4.4.3 VerifyMessageResponse Example Using Redirect Binding

This is an example of a reply to a request to verify that a SAML Message is from a known party and signed according to the configuration for that party, containing the verification response. In this example, the SAML redirect binding was employed.

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequest</a:Action>
    <a:MessageID>urn:uuid:9525a87c-7b8b-474b-9840-eb3f202ab467</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">net.tcp://localhost/samlprotocol</a:To>
  </s:Header>
  <s:Body>
    <msis:VerifyMessageRequest
xmlns:s="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
    <msis:ActivityId>00000000-0000-0000-0000-000000000000</msis:ActivityId>
    <msis:Message>
      <msis:BaseUri>http://contoso.com/</msis:BaseUri>

<msis:SAMLRequest>7b0HYBxJliUmL23Ke39K9UrX4HShCIBgEyTYkEAQ7MGIzeaS7BlpRyMpqyqBymVWzV1mFkDM7Z2
8995777333nvvvfe60510J/ff/z9cZmQBbPbOStrJniGAqsgfP358Hz8iHjFzOlw901638+Wr/Bet86ZnZ55+9tHvv30w
+n0fG+yffDp9P72/v1PH2wf70T59u7s/P755P70/Pze5KP0J/O6KarlZx/tjXc+Ss+aZp2fLZs2W7b00c70w+3dve3dg
zc7u4/u7T7avTd+sLv7Ux+lT6mXYpm1/Oa8bVfNo7t3y2qalfOqar/t39u/d3e5Xhbt3Wf5LK+54cusaYrL/O5H6Um1bH
J0sK6Xj6qsKZpHy2yRN4/a6aPXx188f0S4PJpKo0frZbPKp8V5kc8+St8tymXziEe8+e1VXbXVtCo/OnrMY6r11c0vEYJ
5DVQ/OsKY/CHdfXxX4Bw9vtsn+NH/Aw==</msis:SAMLRequest>

```



```
<msis:RedirectBindingInformation>

<msis:Signature>GdlKRh71Ko9hiCiS2UoDj4fSCplECB0Zu5GGDYlie1lmaMc3zX/EwaIhd+fOZ+NchzJn5rhrEjznI
5KmV3jdtBDgocf2z3C/U/3HeKVde5eqC7NPchGOHhmodt1Ik2KzxmGgOW9st8m4fpLqqrX39vInL9rIfMs3x9IFg3CoC
k=</msis:Signature>
  <msis:SigAlg>http://www.w3.org/2001/04/xmldsig-more#rsa-sha256</msis:SigAlg>

<msis:QueryStringHash>ci5RuRIGSZR2Tz4smxkIL1TU1zqAzYP4Pz798X2Z0cc=</msis:QueryStringHash>
  </msis:RedirectBindingInformation>
  </msis:Message>
  </msis:VerifyMessageRequest>
</s:Body>
</s:Envelope>
```

## 4.5 Logout Operations Examples

### 4.5.1 LogoutRequest Example

This is an example of a message requesting that a SAML logout be performed.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
      s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequest</a:Action>
    <a:MessageID>urn:uuid:17817720-c31e-48e8-8904-067aac199c8d</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">net.tcp://localhost/samlprotocol</a:To>
  </s:Header>
  <s:Body>
    <msis:LogoutRequest
      xmlns:msis="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
      <msis:ActivityId>00000000-0000-0000-0000-000000000000</msis:ActivityId>
      <msis:Message>
        <msis:BaseUri>https://localhost:4343/nunit/FederationPassive</msis:BaseUri>

<msis:SAMLRequest>PHNhbWxwOmxvZ291dFJlcXVlcnQ3SUQ091I184N2YyMmUyNi1mMTcwLTQ4ZDQ0ODEwMS11N2RhODY
xNWJlOWUiIFZlcnNpb249IjIuMCIgSXNzdWVJbnN0Y50PSIyMDA5LTEyLTYeLTE4VDAxOjMxOjIyLjYjc0NVoiIERlc3RpbmF0
aW9uPSJodHRwczovL2xyY2FsaG9zdDo0MzQzL251bm10L0ZlZGVyYXRpb25QYXNzaXZlIiBDb25zZW50PSJcm46b2Fza
XM6bmFtZXM6dGM6U0FNTDoyLjA6Y29uc2VudDp1bnNwZWNPzm1lZCIgTm90T25PckFmdGVyPSIyMDA5LTYeLTE4VDAxOj
M2OjIyLjYjc0NVoiIHhtbG5zOnNhbWxwPSJcm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6ChJvdG9jb2wiPjxJc3N1ZlZI
geG1sbnM9InVybjpvcyYXNpczpuYwllczp0YzptQU1MOjIuMDphc3NlcnRpb24iPmh0dHA6Ly9sb2NhbGhvc3QvPC9Jc3N1
ZlZI+PGRzO1NpZ25hdHVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnIyI+PGRzO1NpZ
251ZEluZm8+PGRzO0kNhbm9uaWNhbG16YXRpb25NZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLz
EwL3htbC1leGtYzE0biMiIC8+PGRzO1NpZ25hdHVyZU1ldGhvZCBBbGdvcm10aG09Imh0dHA6Ly93d3cudzMub3JnLzI
wMDEvMDQveG1sZHNpZy1tb3JlI3JzYS1zaGEyNTYiIC8+PGRzO1JlZmVzZW5jZSBVUkk9IiInF0dmMjJlMjYtZjE3M3M0O
GQ0LTgxMDEtZTdkYTg2MTViZT1lIj48ZHM6VHJhbnNmb3Jtcz48ZHM6VHJhbnNmb3JtIEF5Z29yaXRobT0iaHR0cDovL
3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnI2VudmVsb3B1ZC1zaWduYXR1cmUiIC8+PGRzO1RyYW5zZm9ybSBBbGdvcm
10aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV3Yy1jMTRuIyIgLz48L2RzO1RyYW5zZm9ybSBBbGdvcm
10aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV3Yy1jMTRuIyIgLz48ZHM6
RGlncXN0VmfSdWU+R1pzcGVKUKvN2c2Q09PdG4rZnR4S1RpejgzY9HUTB0VC9jYk2cmNWN08L2RzOkRzP2VzZjZhb
HV1PjwvZHM6UmVmZXJlbnN1PjwvZHM6U2lnbmVksW5mbz48ZHM6U2lnbmF0dXJlVmFsdWU+bEt5ME9QKzc0KzE1bG1RdE
VBOSSxUzRSMjJ3dG9HYW5wN1g4bGhDQU1xeStUbtZ0REIrdC9pQU1mdi9hM3c2dUR3eXVYV11iemV0TW5qUmJFQvDZemF
GZXRwZ3NwSFYaktFVVUxbDjia01WekZiVEVVQjEwRmfNTkdTvjdqRgtSzwTlb21ReGR5Q3FuSFmxcG8rcjEyelRDV3hM
TONJK3pZbk1pMUP6Wxk0PTvwZHM6U2lnbmF0dXJlVmFsdWU+PetteUluZm8geG1sbnM9Imh0dHA6Ly93d3cudzMub3JnL
zIwMDA5MDkveG1sZHNpZyMiPjxkczpYNTA5RGF0YT48ZHM6WU0uOUNlcnRzZmljYXR1Pk1JSU15akNDQVYyZ0F3SUJBZ0
1RUkpLQmZNaHhTYWhDQkJKVC2hsL0EwREF0QmdrcWhraUc5dzBCQVFRrKFEQVNVUk13RUFZRFZRUURFd2xzYjJ0aGJhZ
jMlF3SghjTk1EWXdNVEEwTWpJME5ERTBxAGNOTXpreElqTxhNak0xT1RVNVdqQVNVUk13RUFZRFZRUURFd2xzYjJ0aGJh
aH2jMlF3Z1o4dORRwUPLb1pJaH2jTkFRRUJCUUFEZ1kwQU1UR0pBb0dCQUxqSVJ6U6FdiU1Vpb0k1amdxdKJQblp0Q3luR
ERQNTlsbnNpOW83Q3grK2lUWXEWtzhNqlc0ZjVjS3RSL2doV2JlWm9VVURucl1zSWXhaUU1ncmZic25sNnN2andkQkIwUV
M1Y1NFb0s3T1VqN3hpRnJ2REFNWRhpZcnhCd2VUeZjS3SR1NNZ1hCM3BTZ2tCSHFkCdQbJhd1BwRDVNdUVzTFJSbnJvVh
yMlJpekFnTUJBQUdQU1RCSE1FVUdBVMVvKQVFRK01EeUFFS2tpemh00EExGNkxtbWt3ZXd5V3dUcWhGakFVTvJjD0VBWURW
```

```

UVFERXdsC2IyTmhiR2h2YzNTQ0VFU1NnWHPjY1Vtb1FnUVFwB1pmd05Bd0RRWUpLb1pJaHZjTkFRRUVCUUFZ11FQVBUN
XhBNHc4UHR0SGwvZmtoL1gzVk93R1ZNWUFQYkpNvmdDbDV5TGZRMUxONmdrek4veFVPCpHhVZUY1eXF0Tz1PdXRWRXVFD2
ZzalD4cHlxZVVCt09ZeFVGdDBRS1lvVnNyeDA4M1dETktpbkhiVX1YdkVKLz1Eb2IrdCtpU1BqdTdlbVJKYVRLbm52bWh
HUH10WnU1c2s5c1Q3TzEwSEpTQkJGSEkyRmZkQT08L2RzOlglMD1DZXJ0aWZpY2F0ZT48L2RzOlglMD1EYXRhPjwvS2V5
SW5mbz48L2RzOlNpZ25hdHVyZT48TmFtZU1EIHhtbG5zPSJlcm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6YXNzZXJ0a
W9uIj5iYXI8L05hbWVJRD48c2FtbHA6U2Vzc21vbkluzGV4PjAwMDwvc2FtbHA6U2Vzc21vbkluzGV4Pjwvc2FtbHA6T
9nb3V0UmVxdWVzdD4=</msis:SAMLRequest>
  <msis:PostBindingInformation></msis:PostBindingInformation>
</msis:Message>
<msis:SessionState>http%3a%2f%2flocalhost%2f%2fTrue&aaa&&&&111</msis:SessionState>

<msis:LogoutState>http%3a%2f%2fexternalrp%2fscope?_ID??http%3a%2f%2fexternalrp%2fscope&False&
foo&&&?ID?</msis:LogoutState>
  </msis:LogoutRequest>
</s:Body>
</s:Envelope>

```

## 4.5.2 LogoutResponse Example

This is an example of a reply to a request that a SAML logout be performed, which contains the updated SessionState and LogoutState values.

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
      s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequestResponse</a:Action>
    <a:RelatesTo>urn:uuid:17817720-c31e-48e8-8904-067aac199c8d</a:RelatesTo>
    <a:To s:mustUnderstand="1">http://www.w3.org/2005/08/addressing/anonymous</a:To>
  </s:Header>
  <s:Body>
    <msis:LogoutResponse
      xmlns:msis="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
      <msis:LogoutStatus>InProgress</msis:LogoutStatus>
      <msis:Message>
        <msis:BaseUri>https://localhost:4343/SLO/RedirectResponse</msis:BaseUri>

<msis:SAMLResponse>7b0HYBxJliUmL23Ke39K9UrX4HSHcIBgEYtYkEAQ7MGIzeaS7B1pRyMpqyqBymVWZV1mFkDM7Z
28995777333nvvvfe0510J/ff/z9cZmQBbPbOStrJniGAqsgfP358Hz8iHjfZolw9el5dVov2Vd6sqmWtp2dPP/vo958
9zD69n+/vbM8e7j3Y3t+/v7N9cJDvb0/s3N+f3D+/n+2cTz5KfzKvm6JafvBR3njno/Ssadb52bJps2VLH+3sPNze3dve
PXizs/vo3u6jvb3xg4cPfuqj9GnetMUya/nNeduumkd375bVNCvnVdM+2r+3f+/u6+df3n2Vz4o6n1rMPkpP8APA1/XyU
ZU1RfNomS3y51E7ffT6+IvnjwiPR1Np9Gi9bFb5tDgv8hntjRQ31Q0vIMH53t7+d6n2+e7D3a29w9m+9sHuzu72/mdWX
bw6e79Sf6Qunu3KJfNIybs5i5XddVW06r86OgxEGWVze/1DVNXoMIHx2BCD4N7j6+K3CODIpet1m7bsK/TqpZnv5kVq7
zzf003PrRq/wXrYnweFlRevfo8d0Qrv4ZcsLR/wM=</msis:SAMLResponse>
      <msis:RedirectBindingInformation>

<msis:Signature>AIN+zc9QDY7YZ65zRXz0ob4RMuElAGEPuok37NCdWvubEJ4E3awvi8Ieu+v+LsDhBd+zXZmjb7NDU
XUcoTzqloFNWhlbq34OrMitR4FbGDQMPwByl1Vlmy2MXN7nZvAD+2en+Pd+bkk4P0KMH7PPCQsboj63CyzRfGnV+R81MF
Y=</msis:Signature>
      <msis:SigAlg>http://www.w3.org/2001/04/xmldsig-more#rsa-sha256</msis:SigAlg>
    </msis:RedirectBindingInformation>
  </msis:Message>
  <msis:SessionState>http%3a%2f%2flocalhost%2f%2fTrue&aaa&&&&111</msis:SessionState>

<msis:LogoutState>http%3a%2f%2fexternalrp%2fscope?_ID??http%3a%2f%2fexternalrp%2fscope&False&
foo&&&?ID?</msis:LogoutState>
  </msis:LogoutResponse>
</s:Body>
</s:Envelope>

```

### 4.5.3 LogoutRequest Example - Locally Initiated

This is an example of a message requesting that a SAML logout be performed. In this example, the request is being sent to the endpoint on the local host.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequest</a:Action>
    <a:MessageID>urn:uuid:1fec3465-1008-490d-aeb2-da9b4df4a3d2</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">net.tcp://localhost/samlprotocol</a:To>
  </s:Header>
  <s:Body>
    <msis:LogoutRequest
xmlns:msis="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
      <msis:ActivityId>00000000-0000-0000-0000-000000000000</msis:ActivityId>
      <msis:SessionState></msis:SessionState>
      <msis:LogoutState></msis:LogoutState>
    </msis:LogoutRequest>
  </s:Body>
</s:Envelope>
```

### 4.5.4 LogoutResponse Example:Final Response to Locally Initiated Request

This is an example of a reply to a request that a SAML logout be performed, which contains the updated SessionState and LogoutState values. In this example, the final response to a locally initiated logout request is shown.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequestResponse</a:Action>
    <a:RelatesTo>urn:uuid:1fec3465-1008-490d-aeb2-da9b4df4a3d2</a:RelatesTo>
    <a:To s:mustUnderstand="1">http://www.w3.org/2005/08/addressing/anonymous</a:To>
  </s:Header>
  <s:Body>
    <msis:LogoutResponse
xmlns:msis="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
      <msis:LogoutStatus>LogoutSuccess</msis:LogoutStatus>
      <msis:SessionState></msis:SessionState>
      <msis:LogoutState></msis:LogoutState>
    </msis:LogoutResponse>
  </s:Body>
</s:Envelope>
```

### 4.5.5 LogoutRequest Example with SAMLResponse and RelayState

This is an example of a message requesting that a SAML logout be performed. In this example, the request contains both a SAMLResponse and RelayState.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
```



## 4.5.6 LogoutResponse Example with SAMLRequest and RelayState

This is an example of a reply to a request that a SAML logout be performed, which contains the updated SessionState and LogoutState values. In this example, the response is to a request where the request contains both a SAMLResponse and RelayState.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action
s:mustUnderstand="1">http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/ProcessRequestResponse</a:Action>
    <a:RelatesTo>urn:uuid:9aaa9e19-93b7-46e5-afb8-24804341d435</a:RelatesTo>
    <a:To s:mustUnderstand="1">http://www.w3.org/2005/08/addressing/anonymous</a:To>
  </s:Header>
  <s:Body>
    <msis:LogoutResponse
xmlns:msis="http://schemas.microsoft.com/ws/2009/12/identityserver/samlprotocol/">
      <msis:LogoutStatus>InProgress</msis:LogoutStatus>
      <msis:Message>
        <msis:BaseUri>https://externalrp:4343/SLO/RedirectRequest</msis:BaseUri>
      </msis:Message>
      <msis:SAMLRequest>7b0HYBxJliUmL23Ke39K9UrX4HShCIBgEyTYkEAQ7MGIzeaS7B1pRyMpqqyqBymVWZV1mFkDM7Z2
8995777333nvvvfe60510J/ff/z9cZmQBbPbOStrJniGAqsgfP358Hz8iHjFzOlw9e15dVOv2Vf6L1nnTpm�PP/v0988f
5vd39/J8e+fBwWx7//7+dPvhvXyyPdm�Zg/vH0weTrL7H6U/mddNUS0/+2hvvPNRetY06/xs2bTZsqWPdnYebu/ube8ev
NnZfXRv99HevfG9vYOf+ih9Sr0Uy6z1N+dtu2oe3b2bv2vzepmV9erR/r39e3dfP//y7qt8VtT51GD2UXpSLZscwNf181
GVNUXzaJkt8uZRO330+viL548Ij0dTafRovWxW+bQ4L/LZR+mLqvly+WV9fE699HD71OH2b1Eum0dMls29rOqgraZV+dH
RYx53La9ufilrmrzGuD86wrhp2GULzcp51bR3H98VOEePX9B7Z0/fE955VT2+K28e6bS+zhvMztlylr872tnZeXw38rn5
MOcBo/8H</msis:SAMLRequest>
      <msis:RedirectBindingInformation>
        <msis:RelayState>RelayState</msis:RelayState>
      </msis:RedirectBindingInformation>
      <msis:Signature>TgTFsKkfCEEtm6iul8kZzRzx0OqCxAqelkobQaaS6vV8iXeqmIAdYBvZeTykQaif3KYp5herI6evS
MXA1P7KwX/GG/8o5e6QbNiBZTn48Cti+YJF7yqCZ5HPX/gRg9e9CL8LvMvy8hBa8rDnDOH3eRZFWQNSzJzdVsqS+TNAX+
4=</msis:Signature>
      <msis:SigAlg>http://www.w3.org/2001/04/xmldsig-more#rsa-sha256</msis:SigAlg>
      </msis:RedirectBindingInformation>
    </msis:Message>
    <msis:SessionState></msis:SessionState>
  </s:Body>
</s:Envelope>
<msis:LogoutState>http%3a%2f%2fexternalrp%2fscope?ID??http%3a%2f%2fexternalrp%2fscope&False&f
oo&&&&000?_e9e512ee-078d-454c-93eb-
b1ca958b9ba5?urn%3aoasis%3aname%3atc%3aSAML%3a2.0%3astatus%3aSuccess</msis:LogoutState>
</msis:LogoutResponse>
</s:Body>
</s:Envelope>
```

## 5 Security

### 5.1 Security Considerations for Implementers

Implementers **must have to** ensure that SSL is used to authenticate between clients and servers on different machines, and that the server is the intended server referred to by the server endpoint. Implementers **must also have to** ensure that the remote client role authenticates to the server role such that the server can trust the client to perform SSL client **certificate** authentication where appropriate. Otherwise there are no specific security considerations beyond those specified in normative references.

### 5.2 Index of Security Parameters

None.

## 6 Appendix A: Full WSDL

For ease of implementation, the following example provides the full **Web Services Description Language (WSDL)** ([WSDL]).

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:wsa10="http://www.w3.org/2005/08/addressing"
xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsd/contract"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsd" xmlns:tns="http://tempuri.org/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
targetNamespace="http://tempuri.org/" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types />
  <wsdl:portType name="ISamlProtocolContract" />
  <wsdl:portType name="IAnyActionContract" />
  <wsdl:binding name="DefaultBinding_ISamlProtocolContract" type="tns:ISamlProtocolContract">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  </wsdl:binding>
  <wsdl:binding name="DefaultBinding_IAnyActionContract" type="tns:IAnyActionContract">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  </wsdl:binding>
</wsdl:definitions>
```

## 7 Appendix B: Product Behavior

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

- Windows Server 2003 R2 operating system
- Windows Server 2008 operating system
- Windows Server 2008 R2 operating system
- Active Directory Federation Services (AD FS) 2.0
- Windows Server 2012 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

<1> Section 3.1.3: AD FS 2.0 does use the SOAP endpoint address `net.tcp://localhost/samlprotocol` to establish local connections and the SOAP endpoint address

`https://contoso.com/adfs/services/trust/samlprotocol/proxycertificatetransport`, where `contoso.com` represents the STS server domain name, to establish remote connections.

<2> Section 3.2.3: AD FS 2.0 does use the SOAP endpoint address `net.tcp://localhost/samlprotocol` to establish local connections and the SOAP endpoint address

`https://contoso.com/adfs/services/trust/samlprotocol/proxycertificatetransport`, where `contoso.com` represents the STS server domain name, to establish remote connections.

<3> Section 3.3.3: AD FS 2.0 does use the SOAP endpoint address `net.tcp://localhost/samlprotocol` to establish local connections and the SOAP endpoint address

`https://contoso.com/adfs/services/trust/samlprotocol/proxycertificatetransport`, where `contoso.com` represents the STS server domain name, to establish remote connections.



## 8 Change Tracking

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

## 9 Index

### A

- Abstract data model
  - client (section 3.1.1 23, section 3.3.1 31)
  - server (section 3.1.1 23, section 3.2.1 30)
- Applicability 10
- Attribute groups 22
- Attributes 22

### C

- Capability negotiation 10
- Change tracking 49
- Client
  - abstract data model (section 3.1.1 23, section 3.3.1 31)
  - CreateErrorMessage operation 25
  - initialization (section 3.1.3 23, section 3.3.3 31)
  - Issue operation 25
  - local events (section 3.1.6 30, section 3.3.6 32)
  - Logout operation 25
  - message processing (section 3.1.4 23, section 3.3.4 31)
  - multiple operations 26
  - overview (section 3.1 23, section 3.3 31)
  - sequencing rules (section 3.1.4 23, section 3.3.4 31)
  - SignMessage operation 24
  - timer events (section 3.1.5 30, section 3.3.5 31)
  - timers (section 3.1.2 23, section 3.3.2 31)
  - VerifyMessage operation 24
- Complex types 19
  - overview 19
  - PostBindingType 20
  - PrincipalType 19
  - RedirectBindingType 20
  - RequestType 19
  - ResponseType 19
  - SamlMessageType 20
- CreateErrorMessage operation 25
- CreateErrorMessageRequest example 36
- CreateErrorMessageRequest message 17
- CreateErrorMessageResponse example 37
- CreateErrorMessageResponse message 18

### D

- Data model - abstract
  - client (section 3.1.1 23, section 3.3.1 31)
  - server (section 3.1.1 23, section 3.2.1 30)

### E

- Events
  - local
    - client (section 3.1.6 30, section 3.3.6 32)
    - server (section 3.1.6 30, section 3.2.6 31)
  - local - client 32
  - local - server 31
  - timer
    - client (section 3.1.5 30, section 3.3.5 31)
    - server (section 3.1.5 30, section 3.2.5 31)
  - timer - client 31
  - timer - server 31
- Examples

- CreateErrorMessageRequest 36
- CreateErrorMessageResponse 37
- IssueRequest 33
- IssueResponse 34
- IssueResponse example using artifact binding 35
- LogoutRequest 41
- LogoutRequest example - locally initiated 43
- LogoutRequest example with SAMLResponse and RelayState 43
- LogoutResponse 42
- LogoutResponse example - final response to locally initiated request 43
- LogoutResponse example with SAMLRequest and RelayState 45
- SignMessageRequest 37
- SignMessageResponse 38
- VerifyMessageRequest 39
- VerifyMessageResponse 40
- VerifyMessageResponse example using redirect binding 40

## **F**

- Fields - vendor-extensible 11
- Full WSDL 47

## **G**

- Glossary 7
- Groups 22

## **I**

- Implementer - security considerations 46
- Index of security parameters 46
- Informative references 9
- Initialization
  - client (section 3.1.3 23, section 3.3.3 31)
  - server (section 3.1.3 23, section 3.2.3 31)
- Introduction 7
- Issue operation 25
- IssueRequest example 33
- IssueRequest message 15
- IssueResponse example 34
- IssueResponse example using artifact binding 35
- IssueResponse message 16

## **L**

- Local events
  - client (section 3.1.6 30, section 3.3.6 32)
  - server (section 3.1.6 30, section 3.2.6 31)
- Logout operation 25
- LogoutRequest example 41
- LogoutRequest example - locally initiated 43
- LogoutRequest example with SAMLResponse and RelayState 43
- LogoutRequest message 16
- LogoutResponse example 42
- LogoutResponse example - final response to locally initiated request 43
- LogoutResponse example with SAMLRequest and RelayState 45
- LogoutResponse message 17
- LogoutStatusType simple type 21

## **M**

- Message processing
  - client (section 3.1.4 23, section 3.3.4 31)
  - server (section 3.1.4 23, section 3.2.4 31)

Messages  
  attribute groups 22  
  attributes 22  
  complex types 19  
  CreateErrorMessageRequest 17  
  CreateErrorMessageRequest message 17  
  CreateErrorMessageResponse 18  
  CreateErrorMessageResponse message 18  
  elements 19  
  enumerated (section 2.2.2 12, section 2.2.2.1 13)  
  groups 22  
  IssueRequest 15  
  IssueRequest message 15  
  IssueResponse 16  
  IssueResponse message 16  
  LogoutRequest 16  
  LogoutRequest message 16  
  LogoutResponse 17  
  LogoutResponse message 17  
  LogoutStatusType simple type 21  
  namespaces 12  
  PostBindingType complex type 20  
  PrincipalType complex type 19  
  PrincipalTypes simple type 22  
  RedirectBindingType complex type 20  
  RequestType complex type 19  
  ResponseType complex type 19  
  SamlMessageType complex type 20  
  SignMessageRequest 13  
  SignMessageRequest message 13  
  SignMessageResponse 14  
  SignMessageResponse message 14  
  simple types 21  
  syntax 12  
  transport 12  
  VerifyMessageRequest 14  
  VerifyMessageRequest message 14  
  VerifyMessageResponse 15  
  VerifyMessageResponse message 15  
Multiple operations 26

## **N**

Namespaces 12  
Normative references 8

## **O**

Operations  
  CreateErrorMessage 25  
  Issue 25  
  Logout 25  
  multiple operations 26  
  SignMessage 24  
  VerifyMessage 24  
Overview (synopsis) 9

## **P**

Parameters - security index 46  
PostBindingType complex type 20  
Preconditions 10  
Prerequisites 10  
PrincipalType complex type 19  
PrincipalTypes simple type 22

Product behavior 48

## R

RedirectBindingType complex type 20

References 8

informative 9

normative 8

Relationship to other protocols 10

RequestType complex type 19

ResponseType complex type 19

## S

SamlMessageType complex type 20

Security

implementer considerations 46

parameter index 46

Sequencing rules

client (section 3.1.4 23, section 3.3.4 31)

server (section 3.1.4 23, section 3.2.4 31)

Server

abstract data model (section 3.1.1 23, section 3.2.1 30)

CreateErrorMessage operation 25

initialization (section 3.1.3 23, section 3.2.3 31)

Issue operation 25

local events (section 3.1.6 30, section 3.2.6 31)

Logout operation 25

message processing (section 3.1.4 23, section 3.2.4 31)

multiple operations 26

overview 23

sequencing rules (section 3.1.4 23, section 3.2.4 31)

SignMessage operation 24

timer events (section 3.1.5 30, section 3.2.5 31)

timers (section 3.1.2 23, section 3.2.2 30)

VerifyMessage operation 24

SignMessage operation 24

SignMessageRequest example 37

SignMessageRequest message 13

SignMessageResponse example 38

SignMessageResponse message 14

Simple types 21

LogoutStatusType 21

overview 21

PrincipalTypes 22

Standards assignments 11

Syntax

messages - overview 12

Syntax - messages - overview 12

## T

Timer events

client (section 3.1.5 30, section 3.3.5 31)

server (section 3.1.5 30, section 3.2.5 31)

Timers

client (section 3.1.2 23, section 3.3.2 31)

server (section 3.1.2 23, section 3.2.2 30)

Tracking changes 49

Transport 12

Types

complex 19

simple 21

## V

Vendor-extensible fields 11  
VerifyMessage operation 24  
VerifyMessageRequest example 39  
VerifyMessageRequest message 14  
VerifyMessageResponse example 40  
VerifyMessageResponse example using redirect binding 40  
VerifyMessageResponse message 15  
Versioning 10

## **W**

WSDL 47