

[MS-SAMLPR]: Security Assertion Markup Language (SAML) Proxy Request Signing Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
03/12/2010	1.0	Major	First Release.
04/23/2010	1.0.1	Editorial	Revised and edited the technical content.
06/04/2010	1.0.2	Editorial	Revised and edited the technical content.
07/16/2010	1.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	1.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	1.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	1.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	1.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	1.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
03/25/2011	1.0.2	No change	No changes to the meaning, language, or formatting of the technical content.
05/06/2011	2.0	Major	Significantly changed the technical content.
06/17/2011	3.0	Major	Significantly changed the technical content.
09/23/2011	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
03/30/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/12/2012	3.1	Minor	Clarified the meaning of the technical content.
10/25/2012	3.1	No change	No changes to the meaning, language, or formatting of the technical content.
01/31/2013	3.1	No change	No changes to the meaning, language, or formatting of the technical content.
08/08/2013	3.1	No change	No changes to the meaning, language, or formatting of the technical content.
11/14/2013	3.1	No change	No changes to the meaning, language, or formatting of

Date	Revision History	Revision Class	Comments
			the technical content.

Contents

1 Introduction	7
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	9
1.5 Prerequisites/Preconditions	10
1.6 Applicability Statement	10
1.7 Versioning and Capability Negotiation	10
1.8 Vendor-Extensible Fields	10
1.9 Standards Assignments	10
2 Messages	11
2.1 Transport	11
2.2 Common Message Syntax	11
2.2.1 Namespaces	11
2.2.2 Messages	11
2.2.2.1 SignMessageRequest	12
2.2.2.2 SignMessageResponse	13
2.2.2.3 VerifyMessageRequest	13
2.2.2.4 VerifyMessageResponse	14
2.2.2.5 IssueRequest	14
2.2.2.6 IssueResponse	15
2.2.2.7 LogoutRequest	16
2.2.2.8 LogoutResponse	16
2.2.2.9 CreateErrorMessageRequest	17
2.2.2.10 CreateErrorMessageResponse	18
2.2.3 Elements	18
2.2.4 Complex Types	18
2.2.4.1 RequestType	19
2.2.4.2 ResponseType	19
2.2.4.3 PrincipalType	19
2.2.4.4 SamlMessageType	19
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	21
2.2.6 Attributes	22
2.2.7 Groups	22
2.2.8 Attribute Groups	22
3 Protocol Details	23
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	25
3.1.4.2.1.2	VerifyMessageResponse	25
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	26
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	27
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	31
3.1.6	Other Local Events	31
3.2	Server Details	31
3.2.1	Abstract Data Model	31
3.2.2	Timers	31
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	32
3.3.2	Timers	32
3.3.3	Initialization	32
3.3.4	Message Processing Events and Sequencing Rules	32
3.3.5	Timer Events	32
3.3.6	Other Local Events	32
4	Protocol Examples	33
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	36
4.2	CreateErrorMessage Operation Examples	36
4.2.1	CreateErrorMessageRequest Example	36
4.2.2	CreateErrorMessageResponse Example	37
4.3	SignMessage Operation Examples	38
4.3.1	SignMessageRequest Example	38
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	41
4.5	Logout Operations Examples	42
4.5.1	LogoutRequest Example	42
4.5.2	LogoutResponse Example	43
4.5.3	LogoutRequest Example - Locally Initiated.....	43
4.5.4	LogoutResponse Example:Final Response to Locally Initiated Request.....	44
4.5.5	LogoutRequest Example with SAMLResponse and RelayState	44
4.5.6	LogoutResponse Example with SAMLRequest and RelayState	46
5	Security	47
5.1	Security Considerations for Implementers	47
5.2	Index of Security Parameters	47
6	Appendix A: Full WSDL	48
7	Appendix B: Product Behavior	49
8	Change Tracking.....	50
9	Index	51

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 server.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

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

- certificate**
- SHA-1 hash**
- SOAP**
- SOAP action**
- SOAP body**
- SOAP header**
- SOAP header block**
- SOAP message**
- SOAP mustUnderstand attribute**
- Uniform Resource Locator (URL)**
- Web Services Description Language (WSDL)**
- XML**
- XML namespace**
- XML schema**

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).

SAML: The OASIS Security Assertion Markup Language, as specified in [\[SAMLCore2\]](#) and [\[SamlBinding\]](#).

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

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

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

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

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

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

Security Token Service (STS): A Web service that can issue security tokens, as specified in [\[WSTrust\]](#) section 2.4.

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

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

A reference marked "(Archived)" means that the reference document was either retired and is no longer being maintained or was replaced with a new document that provides current implementation details. We archive our documents online [\[Windows Protocol\]](#).

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

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

If you have any trouble finding [WSAddressing], please check [here](#).

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

[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>

[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.S., 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., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

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. 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 may 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:

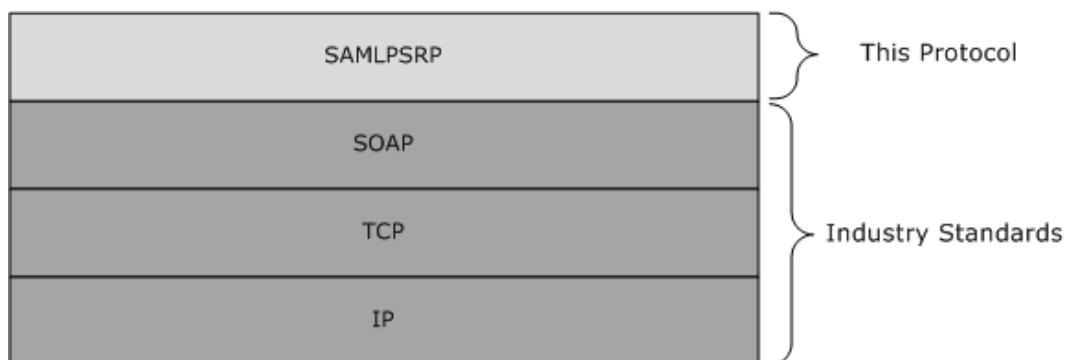


Figure 1: SAMLPR SOAP over TCP layer diagram

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:

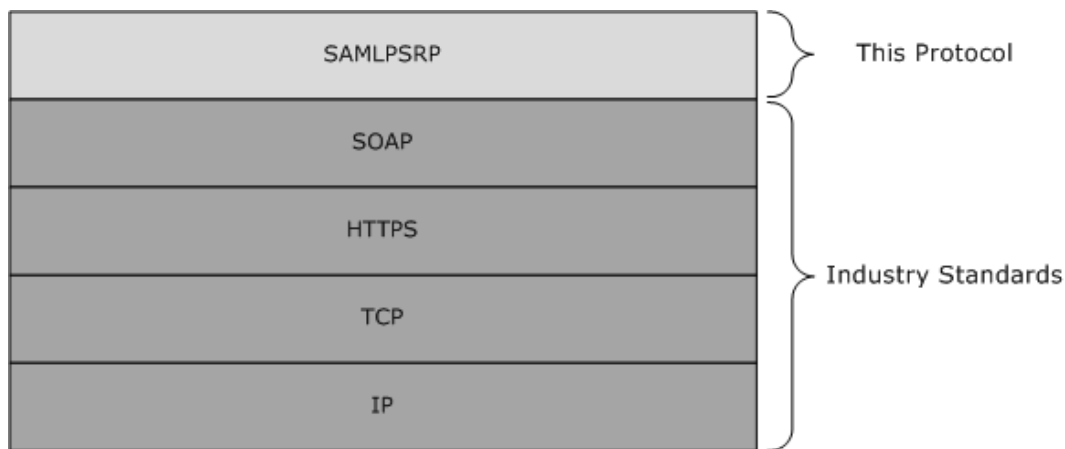


Figure 2: SAMLPR SOAP over HTTPS layer diagram

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

2.2.2 Messages

Message	Description
SignMessageRequest	A message that 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.
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

Message	Description
	party.
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>
```

```

    </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"/>
        <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 schema** 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 SAML post binding .
RedirectBindingType	A structure containing SAML binding information for a SAML redirect binding .

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 be 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

Operation	Description
	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.

Status code	Description
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 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.

Status code	Description
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="samlp: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="samlp: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 be 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.

3.1.6 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>PD94bWwgdmVyc2lvcj0iMS4wIiBlbnVzZGluc2luc2VudD0idXRmLTE2Ij8+PHNhbnVwOkFlldGhuUmVxdWV
zdCBJRd0iX2QzYWNjZWl3LWV1ZjctNDI5Ny1mTgyLWE0NmYxYzQ3NWJjMSIgVmVyc2lvcj0iMi4wIiBjc3N1Zluc3Rh
bnQ9IjIwMDktMTItMThUMDE6MzE6MDYyNDM0WiIgcQ29uc2VudD0idXJuOm9hc2lzMzOm5hbWVzOnRjO1NBTUw6Mi4wOmNvb
nNlbnQ6dW5zcGVjaWZpZWQiIHhtbG5zOnNhbnVwPSJlcm46b2FzaXM6b2FzaXM6dGM6U0FNTDoyLjA6cHJvdG9jb2wiPj
xJc3N1ZXIgeG1sbnM9InVybjpvYXNpczpuYW1lc2p0YzptQU1MOjIuMDphc3NlcnRpb24iPmh0dHA6Ly9leHRlcm5hbHJ
wL3Njb3BlPC9Jc3N1ZXI+PC9zYW1scDpBdXRoblJlcXVlc3Q+</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-
secureconversation/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/PRX3SAAAABK91yIGLJLwXwgu5vEdh3wsm4zf7cBxsK5Waaam5TqQjGDlJ7qhgjnpNBwz9J7r/8fqJLdscGZvU7E
ifqkkoXX0iKdF+fUXxR0oBE/dY4BKGrK1SQ7VgOULAR4Xr39+X8Jp/eeMncIaJuZ01DSB4MwLulVpZKhC3grjfpFAOg
1wBwAAmoP1Iv2HElhlpYbFBmaYmYzpqCOa/Ptr08YCN8YweH1FzEm929H5oEG87TMEjYnuAelBAMGo8BhqBtVS+o16jd
XCSeLF3J/vabemgbxIfJnqh4x5xuYldIRo9FJH78syGjOtGFAVi3KRnpIvnrPp3YKRW0sknIH2LDDzjaFGPZW/w1B0Yen
bWFH+sRkfd+jOqhTVk+3++oeYcZWWSiAZhWDMZKA/kqv3Rh05Drr0v6JbzS3H+PJxzXL1NeEvd8Nhxh+0tINy+I3PWIHX
C7WFgYeS8T1TpaXBq+zrH5DDEQlv4haozU+411T7pBcY9Nd1jLedSK/Eo5/Fyvsm8g2HKL0jgKbr6jB3XYRfFDjAlTIWZ
jq7IZQqqAnaa9xZpcVKx24sHySUo1GK6uFYWpdZYUe5sTmW4bdYVJw6bNS0KEzhekLroce8c9Ldv61idHQJuvG0qT6xU
pCkD6KeUrV6ZQxRoKs5fyemG8sRw7R+p9tLpbPjqnpj4SbAjXwOQJA0ksZ0KCDn+VBQIq/YIc+fWd0Jv6/S+rZLDi1UXg
MYPmdGcfIMFZEMicjknZ8IdcVGIXAuKC9AfffJpfgA+vQOwgqXop6Abi//pKNrDa+ChNOQIkSFQoz5btiOpd63j5dKu/Y5
CqR+tHD7eYsrTf1zdhwi0xYFAn0beoETCRSgsmCgo2iBvWWTxe1rKwPfn5wB0dnznH5ruPAOSQ9alt08k2dqbyavuPri
3MqehLnsBXR9Lh5j45gs19+Injbjv/SelXsbh5BkbTzP9pghkG4ALivz1aRbcQdUDXe85Tb1hcJyDlAVs1PudCMHD1N8p
DDPAkGazcIhTiBnElfljHL7uCeC+UKfEu6Hv8N134yw5vRPErgq8VaRBoUBXVsq9/p4Adv/HGTJbLU2hL1/rr9DqOru3h
Hlp0R18aLddHt/n004awqaIconXGILFqlwMRJXP3J8JL9CncPbp/eszX83o1GILPR+dnSRnAjHQbcGwKSM/5VmtHRVie
g1mXQVcJ90gltdmPictX8luaxruGJezeIzkSwjtXHSg5HPEWg91mtx9Snro4BD9XVeIMHkInBvYczKVlglR+AxopfNCgq
sKyA+EyGVOcsDnqLmnRl2gsvnYlJZBz2uBfgqCq6BLgI0QYnSKDyyEPTfcIn6aupftsh5zmnnd/1vXY6b0TCCXQ+iNuLm
```


5U2VyaWFsTnVtYmVpYjVwZHM6WDUwOUlzc3Vlc1N1cm1hbD48L2RzO1g1MD1EYXRhPjJwvS2V5SW5mbz48ZTpDaXBoZXJEXE
YXRhPjx1OkNpcGhlclZhbHVlPnBVUTQwMmR3cGdUUY9YXWVrK2NvdTAvOGLDYVQ0cDA4NDBTejNCK3RxcmlJWlFCZUFIO
DFzR083NHpSQXRSQ2NVMkova2JBUHBTkZCckdJYWE0eGdGc3NHUUFwWk44RkN6N3pZb2VBNXN1QitGa3pXMOU4Skk3Zi
s2UGxXZGs0OGcrL1p3d1lKd1poTDhZWTJQT3ZYV1NzRzJMUGoYrHpkdFp0eHJVtU9kRT08L2U6Q21waGvYVmFsdWU+PC9
1OkNpcGhlclZhbHVlPC91OkVtY3J5cHRlZEtlet4R0L2leUluZm8+PHh1bmM6Q21waGvYrGF0Y248eGvYzPdaXBoZXJEXE
YXWx1Z25CMQzN3FBUWJwEURTeTJ5c3Joa3ZiWk0zcme1dk0vbjVuUUFhREFNcVdrWFhCdm9DTEYrdjNwEYyVDRoWGJ3U
1kvOUpVZEpJNDJwMXVFTGpxd3NXRmxNK1QwU2NJeDQ3ZG8Wz20yUCtFaG40amlXZTzWcGx4dUcWVFFpeWnkZE1OZEKzTT
d4UnNLVEhHVWc2dnNiN294bnBoWFF4TXd2SVB1WgyzQW10ZEx3YXN2RUEyMGg2N3JwS1RFRm11QU9WVksvTEFvNXd2eTZ
MZHpiCzRLMEVpVW5NT09sdGREN1pKUT1Sc1pyMHZEOGM2K2EybdBNYUNSL0pIWGJpTmlraGtmclFCWThqc1FFRHQ2VBoz
ZENXUEJtNgG2c1FxoQ51V2NDW1Jzc1BZYk5Gek1GTHhVsnJVRUVHMkJBOWP5a0x3UkhtSVUxRFZ0cmY0a3Vrbk01TkhNb
UMxU2JFQ2tqTDY3emRHOHgzSkeydl2bnhKUUQxTH1YcmZrd2VCRU90c1dJT3BCCwVmeStnMXVQLy9QsK02ZHZBSGU5az
1vS2JQemJ5UWQ1SVRiY1lZSx1pVFBKZ0UrNEkralIyT211eWVHem1zY0hZc2s3MG5wRWxGb1Rk2NXZXZyb3BTd28yRnZ
jNVF0V3dicHN4UnBXS3E4OCTjcxPuV0x0S01zMG92Y2ZjNzV1aWFnM2xpK2NRajVE5m5G1pSenpJmzFo5UpaRFJ3OXpM
TmR6eU8zcn1J3RmFwR3hESTF6VDdwdF1xSDJKV3ZnZM1nb29rSWH26ZDFXaEFDShNNnes4Q09nWm9rsvSNWLT
XJYWmpwQkhXV1AxZG1pb3JVZ0hZa3czY0xkUzF4bTc5Rk9Mz21JbWRMcmhSRFFza0VxeW1Rc1g5M2FBVHBTanZvRegzMn
RsNGLZc2tnY1MvOFJKaGRHMnUxUXJ4dX1sXQ1MmrdBQvRwPxa1pZRHFXcONQY1JxYWFVXTNybGJrU1OT29sL2JvbU1
oenRFZi81dW16UTFvY1AwV2JOUFVneXNPTnhtY29HQ3VIS0xzcDBuRwKxdXhMDFN5R0RyQTJKEChOrNpHb0hraFl6Q1QY
WWhEM1ZmQ2x5YXR0N1hEOXRvT29qZUJ3bjYyWmxqMWhIxDmFNJRVV3VfHmZnoxbmlv1d1YIYUvR0v1Z1V2Nack140XkIQ
TNZMGt2L3RJETelqldrL1dUTEM2dlRRcEl6NXkzT1cyazNPdUZJTYtmRWRCYT1XTNnb05HSENOwEs1bEsxTDBmTgdYRG
NrYwxiTXNtNzRhaHE3L2xwZmJyU1F1cGdZY3ZCUNJsTmVbEhGam150F3Mv1RUHNBtm9FWkNt2VVeCtJ0a1RqS010eXh
kdEZTnitKbTBJZ2JPb3FueGc3ejJKc1dwYVfocmtDcHg1LzFXbn11ZVRaMwJmV3cyaXZ4N3hnUjMxZXJ1NUFTTGIdUtH
U211LenVzFA4NHFwR3hEOHA4Y0ExQmhoNzVXmVFBMG9Sc0o1T2d5OVc3VXQ2dWY5R1hjcnhxj1LLNzR1WnhJMDbVtVNSVz
lgzMX05eW9pbW14WHNBZGtWR0p4bD1jc0ROeTdOcE1iOXBoZmpKTGFTSEFhYjhcQcJ14U0h1NUU0L2ZrcVNXSk9kt2tJcX
JjUF24TV1Lb0pvaE9YTU84Ym1INnhvdGZHDZRE11cW04ZS5tKvYnjV6UHg4cTBxVHY1b1JvQUFFV1YyUUVsc3daU2d
uTk1OSkRvVwXZMGd4RkNxaWduN05Kc0Q3WVYyVWZhdnhtOVJmV2hGbHdON0czMwD1Vk05R1Nrd0pEa3BLZVpsUU5tVHdm
WTZjVfUx4aJzT1ovQ3hH2IwYmKl3hNz42MG14Y1V1dEtYVUhhkbjRvaG9nUXviUUpHM21TTLFgS1JZp0G1E0E0Z5T0Rmx3M
GFwd3pCYXyWt0tkU3NCQjAzUGN0YjVGVWZuNm5iNXVpTtnaTW1YbFpOWsvaVdRbnVQT1Vya2hZaVpLNFR3SDBVdU4rc3
V5YmNKNyt2dmlwenh1MkxLZjF3YisZajNwCXJtZn1BYjVzUW430FpUSF1OUFpFYXU4RH1Ya0E0cEY3MH12SnVRL2tGS31
4WTVUHTA4NnTbHRQekhaN2ZkU2dGVFRLTE5IT09BeTi3SnJiVWFek2RbDmYramY5TVA3M31BajZnVmRzRXBETzNudjJ4
dXRiNHpCNktxSYRkBNzK3ZkNEJpQWDFLbE8xv0tXOUFRN0FaSkFzSTJQMzZh0W9PYVJwTkpITS9yaGJGK1VZcFMvY3pvW
Ut5NwVYwC9wb2pxellpR0s1VHg1OExPdjMzMR3ZwP5aGkzNm95NEg3K1NwWDJRYW1PNGVYTW1R21SQ01vd0V6RTFLQV
lCOE9hY1NRSER4U2JVLyTtUG1aNE9QZk5hMV2RFJCbU43cVd3Tk1LSh1Rc0FDaMwWNE9YUW1CeW0zRE1weWx2SWUva2Z
RckfKbHBRnIza0V3RU0xUit1Mk12R1Z4NEFTdl1dWHRsR1NUdW1UWk1udzRmajhXUf1RdFk1SFzhVxd2dW1DTMRRL21a
WmV1dUx4aJzT1ovQ3hH2IwYmKl3hNz42MG14Y1V1dEtYVUhhkbjRvaG9nUXviUUpHM21TTLFgS1JZp0G1E0E0Z5T0Rmx3M
3hwt2JkamhHTHNYaTZqR2RWTytMbTRKS0R2eTQreVRwa3F0K3JwTgD5M1VSSFJ3SERUWVNQC2NraG04TVAvbmswT1ZLK2
JiN2pRZDd4bXBoNFvRmXV2aStvcGZSS01FTVo3WjVubFgvSkpWOC9DRXk1dTVWmi96ZjVXZjR6eHpmSWExblp0WkxKVUZ
oMnFoZxJ1LzNEZjMyU1A30V1UblpDdTz1WkU5RVVRMFRjUG1PdTBdVh04cVM0VUpCL0tqQWV6Q3ZJQ0I2dn0VndSOWh6
Ti9vazdVK2NLUHRsK1M3WfNRZDdpN0didFI5S1c2b1ZPQW1yVG05RGM5dTB1bzFuNXpPTXpKYUtUMLdn0EvgOVJOSTE0c
nAyQ3NpVDgyQnhFTnRyMXA3L1BtNndCQ1FBbws5RU50aTdDcFNzDHRuOWpzaJjGwMnkd0o1REZIRGQzRWZveGRLMkw0Wk
5kdE9pSUpaeEQ5bHV1QytSNHczT2V3b3pQWENVd1NjenBiRXQyUHNpDnR6VzFFVEYwZU5xcEt1R2FsSEhOUkRML0x6UGp
KYzd1S1BmUXZ5bHp0TksZTet1bm9rMkFTUUYcVhCUGpWWDNdandOemdOWFdadixUC9ue11UakpuUUDJVFdBmitqNGLR
VWPCSFaxVfZ3K3EVDUwCtsQ215Qys4Rm11T3Y2R0JNTG2IX3pLeVfXV5FVmpORytLL2dxWEPsBhLWM1Zp0G1E0E0Z5T0Rmx3M
zhmNt1Lem1CdJFHU1VsYnBaY1BkVhJWSHcRcHNYRjNBW1RVU1d0TmPFMhPrY2pMUUVuclhTRHN6S0dzVmh2d1BOQjhxL2
VvNnE0dTIzNnBGBE5tN0hzRjB3UHfWl0xtQVNZR1RNUDBkQitYMG1FTytLcWgXzqhgqkxkWWFVmjJrdCtGcStCbFhSG1
MTSs0QlqgvNnBhU2VsaU1PwkhjRkFsvEYxUU5iak1SaXdhbnJDQ1JpcHg4b1ZuVedObWF2RzFVRnQzcnBucGFCR1pnMnNW
TWpha2kzKpXUkxkbkVxejBLdHBUWFRaWURjK3I3U1M2TE02K1dXZitpTTR2Qkx1Ly9tejN0SD1haFpob0s1c0NEdC9uz
GJHSUk4emtYbngvai9aUXdtek9vMndwSFJmbXNYS25UMmNRNhp4dU51TzVUVVZTTmN1cko5dmhidTR3OTc2b2R4K0JJK2
1QZwtzL0hxTmxibXhYymx2cz11cTVXWVmUmhIVjFpd3NITWNKYThnZGdjY1LWN0NSaGRxUitPeVZsvWFtBepSemFEYtd
SWX1TbU1wbzZaT3BUeTdhQWVNaWhnNit4VfHMRVBDZHZCQjczEduRumJIRUpQbmJocEpXUFFkc21uWk1jYkZjREF5aFhQ
VmdQVXJGYNi0VVRlBDFYyVdPcDNGUS94c1BHWGFVSmZRN0ZRY1ZJNkdoUEFBamVpK1Y1YW5Wa3NpekpTY0dsNXp5OE1GQ
1E1eEFSQ1pCSFJ5LytNMWtvNzNNUnExS8yZzhrR0k1N3gyQWczZ1hQsZJvRHVVTKdWvEc5c2JZRMv5VGxCTVBuRmtaTG
dLT1BhcZLZGeyUmNWZ02Q0tGeCt3bGU1T1Q5TTQ1d1ZGbzBLVEV3UF16QmdTa24zS2NqN0t5NmRUVGxIaUtjU0QxaCt
Pd2FyM2dZ31wcGcrMDBDSG10Z2NOU282cmoxV1V6cmg5STJvQkRQMGJBUc9BS2Mva0MycKriVmRHTGpTbS9FSDducTnw
OULzeKhhL1YrOVdFMNrvZklqL1JNRY8venV2T08L3h1bmM6Q21waGvYVmFsdWU+PC94Zw5jOkNpcGhlclZhbHVlPC94Zw5j
W5jOkVtY3J5cHRlZERhdGE+PC9FbMnYexB0ZWRBc3NlcmRpb24+PC9zYW1scDpsZXNwb25zZT4=/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/ProcessRequestResponse</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/CTgpb07ZhNzAgEAN1B90ECfpNEVLg=</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/ProcessRequest</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>PD94bWwgdmVyc2lvcj0iMS4wIiBlbmNvZGluc2luc30idXRmLTE2Ij8+PHNhbnVwOkF1dGhuUmVxdWV
zdBjJRD0iXzIwN2Y2dGhLTA1Y2gtNGMzOS1iMTE0LTgyYzcs5ZTk1Y2NmOCgVmVyc2lvcj0iMi4wIiBJc3N1ZU1uc3Rh
```

```

bnQ9IjIwMDktMtItMThUMDE6MzE6MTEuODYzWiIgcQ29uc2VudD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOmNvb
nNlbnQ6dW5zcGVjaWZpZWQ6IFByb3RvY29sQmluZGluZz0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOmJpbmRpbm
dzOkhUVFAtUmVkaXJlY3QiIHhtbG5zOnNhbwXwPSJlcm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6cHJvdG9jb2wiPjx
Jc3N1ZXIgeGlsbnM9InVybjpvYXNpczpuYW1lc2p0YzpzTU1MOjIuMDphc3N1cnRpb24iPmh0dHA6Ly9leHRlcm5hbHJw
L3Njb3BlPC9Jc3N1ZXI+PC9zYW1scDpBdXRob1JlcXVlc3Q+</msis:SAMLRequest>
  <msis:PostBindingInformation></msis:PostBindingInformation>
</msis:Message>
</msis:CreateErrorMessageRequest>
  <msis: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/Proc
essRequestResponse</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
28995777333nvvffe60510J/ff/z9cZmQBbPbOStrJniGAqsgfP358Hz8iHjfZolw9epU3q2rz50nZ088++v2zLL+/vzO
5tz3Zy/e29/PJ/e1sZ7q7ne8/yPbuP9g9mB3MPkp/Mq+bolp+9tHeeOej9Kxp1vnZsmmzZU5f7ew83N7d2949eLOz++je
7qPd3fHDvb2f+ih9mjdtsxafnPetqvm0d27+bs2r5dZWa/ufpSeABEAwdfLR1XWFM2jZbbIm0ft9Nhr4y+eP6L+Hk210
aP1sln10+K8yAmjs6UZx5uKhrG38yD/NHUbQbe/czw6296f3Hm5Pdnf3tw/2pg8e5g/vT6fnBx+17xblsnnEZnjc5aqu2m
palR8dPebBlvLq5peypslrDPajIwyWxlpW06ycV0179/FdgXOkk/C6zdp1E/51Us3y9Cezcplv7qfhljqPs7z+6IPAVKi
O1+18STPR0tr81N49eny3C67zKf3TMMHR/wM=</msis:SAMLResponse>
        <msis:RedirectBindingInformation>
        <msis:Signature>R1FtupsaiITbNa5wL4+mOnuFpRBYs5kq/ni5ycqNprqpol0c5+RUOA5/8RkMY787oB8l7FfFJOYw
3FkIhWayaPqclb1HFp7AcuJFPmWVT2bGXbdRV6sCFV0g5X01PsYG+a/9EZdiYUaMCRUvOds0s5SdtmL95FCQpLxkG5PEk
w=</msis:Signature>
          <msis:SigAlg>http://www.w3.org/2001/04/xmldsig-more#rsa-sha256</msis:SigAlg>
        </msis:RedirectBindingInformation>
      </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>
      <msis:Message>
        <msis:BaseUri>http://contoso.com/</msis:BaseUri>

<msis:SAMLRequest>PHNhbWxwOkF1dGhuUmVxdWVzdCBJRd0iXzA4MTZjZjJiLTg2YzUtNDU2Ny04MGV1LTFlkZjVmYjV
jZmYzYiIgmVyc2lrbj0iMi4wIiBjc3N1ZU1uc3RhbnQ9IjIwMDktMTItMTUUMDE6MzE6MTMuNTEzWiIgrGVzdgLUYXRp
b249Imh0dHBzOi8vbG9jYXob3N00jQzNDMvbnVuaXQvRmVkJXJhdGlvb1Bhc3NpdmUvIiBDb25zZW50PSJlcm46b2Fza
XM6bmFtZXM6dGM6U0FNTDoyLjA6Y29uc2VudDplbnNwZWNPZml1ZCIgeG1sbnM6c2FtbHA9InVyb2pvczpuYWllcz
p0YzptQU1MOjIuMDpwcm90b2NvbCIgLz4=</msis:SAMLRequest>
        <msis:PostBindingInformation></msis:PostBindingInformation>
      </msis:Message>
      <msis:Principal>
        <msis:Type>Scope</msis:Type>
        <msis:Identifier>http://externalrp/rp1</msis:Identifier>
      </msis:Principal>
    </msis:SignMessageRequest>
  </s:Body>
</s:Envelope>
```

4.3.2 SignMessageResponse Example

This is an example of a reply to a request to create a signed SAML Message, which contains the resulting 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/ProcessRequestResponse</a:Action>
    <a:RelatesTo>urn:uuid:5654c3f9-691f-4f9e-aa51-d5d37060dc88</a:RelatesTo>
    <a:To s:mustUnderstand="1">http://www.w3.org/2005/08/addressing/anonymous</a:To>
  </s:Header>
  <s:Body>
```



```

    <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>7b0HYBxJliUmL23Ke39K9UrX4HShCIBgEyTYkEAQ7MGIzeaS7B1pRyMpqqyqBymVWZV1mFkDM7Z2
8995777333nvvvfe60510J/ff/z9cZmQBbPbOStrJniGAgsgfP358Hz8iHjfZolw901638+Wr/Bet86ZNz55+9tHvv30w
++n0fG+yffDp9P72/v1PH2wf7OT59u7s/P755P70/Pze5KP0J/O6KarlZx/tjXc+Ss+aZp2fLZs2W7b00c70w+3dve3dg
zc7u4/u7T7avTd+sLv7Ux+lT6mXypm1/Oa8bVfNo7t3y2qalfOqar/t39u/d3e5Xhbt3Wf5LK+54cusaYrL/O5H6Um1bH
J0sK6Xj6qsKZpHy2yRN4/a6aPXX188f0S4PJPko0frZbPKp8V5kc8+St8tymXziEe8+elVXbXVtCo/OnrMY6r11c0vEYJ
5DVQ/OsKY/CHdfXxX4Bw9vtsn+NH/Aw==</msis:SAMLRequest>

          <msis:RedirectBindingInformation>

          <msis:Signature>GdlKRh71Ko9hiCiS2UoDJ4fSCp1eCB0Zu5GGDYlie1lmaMc3zX/EwaIHd+fOZ+NchzJn5rhrEjznI
5KmV3jdtBDgocf2z3C/U/3HeKVde5eqC7NPchGOHhmodt1Ik2KzxmMgOW9st8m4fpLqqrX39oVInL9rIfMs3x9IFg3CoC
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>

```



```

<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>7b0HYBxJliUmL23Ke39K9UrX4HShCIBgEyTYkEAQ7MGIzeaS7B1pRyMpqqqBymVWZV1mFkDM7Z
28995777333nvvvfe6O510J/ff/z9cZmQBbPbOStrJniGAqsgfP358Hz8iHjfZolw9e15dV0v2Vd6sqmWtp2dPP/vo958
9zD69n+/vbM8e7j3Y3t+/v7N9cJDvbO/s3N+f3D+/n+2cTz5KfzKvm6JafvbR3njno/Ssadb52bJps2VLH+3sPNze3dve
PXizs/vo3u6jvb3xg4cPfuqj9GnetMUya/nNeduumkd375bVNCvnVdM+2r+3f+/u6+df3n2Vz4o6n1rMPkpP8APA1/XyU
ZU1RfNomS3y51E7ffT6+IvnjwiPR1Np9Gi9bFb5tDgv8hnhhtjRQ31Q0vIMH53t7+d6n2+e7D3a29w9m+9sHuzu72/mDWX
bw6e79Sf6Qunu3KJfNIybS5i5XddVW06r86OgxEGWVze/1DVNXoMIHx2BCD4N7j6+K3C0dIpet1m7bsK/TqpZnv5kVq7
zzf003PrRq/wXrYnweFlRevfo8d0Qrv4ZcsLR/wM=</msis:SAMLResponse>
        <msis:RedirectBindingInformation>

<msis:Signature>AIN+zc9QDY7YZ65zRXz0ob4RMuElAGEPuok37NCdWvubEJ4E3awvi8Ieu+v+LsDhBd+zXZmjb7NDU
XUcoTzql0FNoWhlbq34OrMitR4FbGDQMpwBy1Vlmy2MXN7nZvAD+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&True&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>
```

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:SAMLRequest>7b0HYBxJliUmL23Ke39K9UrX4HShCIBgEyTYkEAQ7MGIzeaS7B1pRyMpqqyqBymVWZV1mFkDM7Z2
8995777333nvvvfe60510J/ff/z9cZmQBbPbOStrJniGAqsgfP358Hz8iHjfZolw9e15dVOv2Vf6L1nnTpmdPP/vo988f
5vd39/J8e+fBwWx7//7+dPvhvXyyPdmdZg/vH0weTrL7H6U/mddNUS0/+2hvvPNRetY06/xs2bTZsqWPdnYebu/ube8ev
NnZfXRv99HevfG9vYOf+ih9Sr0Uy6z1N+dtu2oe3b2bv2vzepamV9erR/r39e3dfP//y7qt8VtT51GD2UXpSLZscwNf18l
GVNUXzaJkt8uZR0330+viL548Ij0dTafRovWxw+bQ4L/LZR+mLqv1y+WV9fE699HD710H2b1Eum0dMls29rOqqraZV+dH
RYx53La9ufilrmrzGuD86wrhp2GU1zcp51bR3H98VOEePX9B7Z0/fE955VT2+K28e6bS+zhvMztlylr872tnZeXw3rn5
MOCB0/8H</msis:SAMLRequest>
      <msis:RedirectBindingInformation>
        <msis:RelayState>RelayState</msis:RelayState>

<msis:Signature>TgTFsKkfCEEtmiu18kZzRzx00qCxAqelkobQaaS6vV8iXeqmIAdYBvZeTykQaif3KYp5herI6evS
MXAlP7KwX/GG/8o5e6QbNiBZTn48Cti+YJF7yqCZ5HPX/gRg9e9CL8LvMvy8hBa8rDnDOH3eRZFwQNSzJzdVVSqs+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>

<msis:LogoutState>http%3a%2f%2fexternalrp%2fscope?ID??http%3a%2f%2fexternalrp%2fscope&False&f
oo&&&&000?_e9e512ee-078d-454c-93eb-
b1ca958b9ba5?urn%3aoasis%3anames%3atc%3aSAML%3a2.0%3astatus%3aSuccess</msis:LogoutState>
  </msis:LogoutResponse>
</s:Body>
</s:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

Implementers must 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 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:misc="http://schemas.microsoft.com/ws/2005/12/wsdl/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/wsdl" 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](#) 32)
 server ([section 3.1.1](#) 23, [section 3.2.1](#) 31)
[Applicability](#) 10
[Attribute groups](#) 22
[Attributes](#) 22

C

[Capability negotiation](#) 10
[Change tracking](#) 50
Client
 abstract data model ([section 3.1.1](#) 23, [section 3.3.1](#) 32)
 [CreateErrorMessage operation](#) 26
 initialization ([section 3.1.3](#) 23, [section 3.3.3](#) 32)
 [Issue operation](#) 25
 local events ([section 3.1.6](#) 31, [section 3.3.6](#) 32)
 [Logout operation](#) 25
 message processing ([section 3.1.4](#) 23, [section 3.3.4](#) 32)
 [multiple operations](#) 26
 overview ([section 3.1](#) 23, [section 3.3](#) 31)
 sequencing rules ([section 3.1.4](#) 23, [section 3.3.4](#) 32)
 [SignInMessage operation](#) 24
 timer events ([section 3.1.5](#) 31, [section 3.3.5](#) 32)
 timers ([section 3.1.2](#) 23, [section 3.3.2](#) 32)
 [VerifyMessage operation](#) 24

Complex types

[overview](#) 18
[PostBindingType](#) 20
[PrincipalType](#) 19
[RedirectBindingType](#) 20
[RequestType](#) 19
[ResponseType](#) 19
[SamlMessageType](#) 19
[CreateErrorMessage operation](#) 26
[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](#) 32)
 server ([section 3.1.1](#) 23, [section 3.2.1](#) 31)

E

Events
 local
 client ([section 3.1.6](#) 31, [section 3.3.6](#) 32)
 server ([section 3.1.6](#) 31, [section 3.2.6](#) 31)
 timer
 client ([section 3.1.5](#) 31, [section 3.3.5](#) 32)

 server ([section 3.1.5](#) 31, [section 3.2.5](#) 31)

Examples

[CreateErrorMessageRequest](#) 36
[CreateErrorMessageResponse](#) 37
[IssueRequest](#) 33
[IssueResponse](#) 34
[IssueResponse example using artifact binding](#) 36
[LogoutRequest](#) 42
[LogoutRequest example - locally initiated](#) 43
[LogoutRequest example with SAMLResponse and RelayState](#) 44
[LogoutResponse](#) 43
[LogoutResponse example - final response to locally initiated request](#) 44
[LogoutResponse example with SAMLRequest and RelayState](#) 46
[SignInMessageRequest](#) 38
[SignInMessageResponse](#) 38
[VerifyMessageRequest](#) 39
[VerifyMessageResponse](#) 40
[VerifyMessageResponse example using redirect binding](#) 41

F

[Fields - vendor-extensible](#) 10
[Full WSDL](#) 48

G

[Glossary](#) 7
[Groups](#) 22

I

[Implementer - security considerations](#) 47
[Index of security parameters](#) 47
[Informative references](#) 9
Initialization
 client ([section 3.1.3](#) 23, [section 3.3.3](#) 32)
 server ([section 3.1.3](#) 23, [section 3.2.3](#) 31)
[Introduction](#) 7
[Issue operation](#) 25
[IssueRequest example](#) 33
[IssueRequest message](#) 14
[IssueResponse example](#) 34
[IssueResponse example using artifact binding](#) 36
[IssueResponse message](#) 15

L

Local events
 client ([section 3.1.6](#) 31, [section 3.3.6](#) 32)
 server ([section 3.1.6](#) 31, [section 3.2.6](#) 31)
[Logout operation](#) 25
[LogoutRequest example](#) 42
[LogoutRequest example - locally initiated](#) 43
[LogoutRequest example with SAMLResponse and RelayState](#) 44

[LogoutRequest message](#) 16
[LogoutResponse example](#) 43
[LogoutResponse example - final response to locally initiated request](#) 44
[LogoutResponse example with SAMLRequest and RelayState](#) 46
[LogoutResponse message](#) 16
[LogoutStatusType simple type](#) 21

M

Message processing
 client ([section 3.1.4](#) 23, [section 3.3.4](#) 32)
 server ([section 3.1.4](#) 23, [section 3.2.4](#) 31)

Messages
 [attribute groups](#) 22
 [attributes](#) 22
 [complex types](#) 18
 [CreateErrorMessageRequest message](#) 17
 [CreateErrorMessageResponse message](#) 18
 [elements](#) 18
 [enumerated](#) 11
 [groups](#) 22
 [IssueRequest message](#) 14
 [IssueResponse message](#) 15
 [LogoutRequest message](#) 16
 [LogoutResponse message](#) 16
 [LogoutStatusType simple type](#) 21
 [namespaces](#) 11
 [PostBindingType complex type](#) 20
 [PrincipalType complex type](#) 19
 [PrincipalTypes simple type](#) 21
 [RedirectBindingType complex type](#) 20
 [RequestType complex type](#) 19
 [ResponseType complex type](#) 19
 [SamlMessageType complex type](#) 19
 [SignMessageRequest message](#) 12
 [SignMessageResponse message](#) 13
 [simple types](#) 21
 [syntax](#) 11
 [transport](#) 11
 [VerifyMessageRequest message](#) 13
 [VerifyMessageResponse message](#) 14

[Multiple operations](#) 26

N

[Namespaces](#) 11
[Normative references](#) 8

O

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

P

[Parameters - security index](#) 47
[PostBindingType complex type](#) 20
[Preconditions](#) 10
[Prerequisites](#) 10
[PrincipalType complex type](#) 19
[PrincipalTypes simple type](#) 21
[Product behavior](#) 49

R

[RedirectBindingType complex type](#) 20

References
 [informative](#) 9
 [normative](#) 8

[Relationship to other protocols](#) 9
[RequestType complex type](#) 19
[ResponseType complex type](#) 19

S

[SamlMessageType complex type](#) 19

Security
 [implementer considerations](#) 47
 [parameter index](#) 47

Sequencing rules
 client ([section 3.1.4](#) 23, [section 3.3.4](#) 32)
 server ([section 3.1.4](#) 23, [section 3.2.4](#) 31)

Server
 abstract data model ([section 3.1.1](#) 23, [section 3.2.1](#) 31)
 [CreateErrorMessage operation](#) 26
 initialization ([section 3.1.3](#) 23, [section 3.2.3](#) 31)
 [Issue operation](#) 25
 local events ([section 3.1.6](#) 31, [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](#) 31, [section 3.2.5](#) 31)
 timers ([section 3.1.2](#) 23, [section 3.2.2](#) 31)
 [VerifyMessage operation](#) 24
 [SignMessage operation](#) 24
 [SignMessageRequest example](#) 38
 [SignMessageRequest message](#) 12
 [SignMessageResponse example](#) 38
 [SignMessageResponse message](#) 13

Simple types
 [LogoutStatusType](#) 21
 [overview](#) 21
 [PrincipalTypes](#) 21

[Standards assignments](#) 10
[Syntax - messages - overview](#) 11

T

Timer events
 client ([section 3.1.5](#) 31, [section 3.3.5](#) 32)
 server ([section 3.1.5](#) 31, [section 3.2.5](#) 31)

Timers
 client ([section 3.1.2](#) 23, [section 3.3.2](#) 32)
 server ([section 3.1.2](#) 23, [section 3.2.2](#) 31)
[Tracking changes](#) 50
[Transport](#) 11
Types
 [complex](#) 18
 [simple](#) 21

V

[Vendor-extensible fields](#) 10
[VerifyMessage operation](#) 24
[VerifyMessageRequest example](#) 39
[VerifyMessageRequest message](#) 13
[VerifyMessageResponse example](#) 40
[VerifyMessageResponse example using redirect binding](#) 41
[VerifyMessageResponse message](#) 14
[Versioning](#) 10

W

[WSDL](#) 48