DRUMMOND GROUP INC.
TEST PLAN FOR LIBERTY
ALLIANCE SAML TEST EVENT
TEST CRITERIA

SAML 2.0

VERSION 3.0 ERRATA J

Prepared & Facilitated By:
DRUMMOND GROUP INC.
www.drummondgroup.com
Contents

www.drummondgroup.com ............................................................................................................... i
Contents ........................................................................................................................................ iii
Contents ........................................................................................................................................ iii
Test Plan Amendments ................................................................................................................ 1
Introduction ................................................................................................................................... 2
Overview of Test Plan ...................................................................................................................... 2
Test Plan History ............................................................................................................................. 2
SAML Conformance Modes ........................................................................................................... 2
GSA Profile .................................................................................................................................... 3
Technical Requirements ................................................................................................................. 4
Metadata ..................................................................................................................................... 4
IdP Authentication ......................................................................................................................... 4
Trivial Processing .......................................................................................................................... 4
Authentication Contexts ................................................................................................................ 4
Name Identifier Formats .............................................................................................................. 5
XML Signatures ............................................................................................................................. 5
XML Encryption ............................................................................................................................ 6
Attribute Profiles ......................................................................................................................... 6
Test Cases .................................................................................................................................... 7
Overview of Test Case Description ............................................................................................... 7
Test Cases Associated with Conformance Modes ......................................................................... 7
General Test Case Requirements .................................................................................................. 8
Test Case A – Redirect Binding ...................................................................................................... 8
Test Case B – SOAP Binding ........................................................................................................... 11
Test Case C – POST Binding ......................................................................................................... 14
Test Case D – Extended SAML Modes ........................................................................................ 17
Test Case E – IDP Introduction .................................................................................................... 20
Test Case F – Single Session Logout ............................................................................................ 21
Test Case G – Unsolicited <Response> ....................................................................................... 23
Test Case H – Affiliations .............................................................................................................. 24
Test Case I – ECP ........................................................................................................................ 26
Test Case J – SAML Authentication Authority ............................................................................. 27
Test Case K – SAML Attribute Authority ..................................................................................... 29
Test Case L – SAML Authorization Decision Authority ................................................................. 31
Test Case M – SAML URI Binding ................................................................................................. 33
Test Case N – Error Testing .......................................................................................................... 34
Test Case O – GSA Profile ............................................................................................................. 36
References ................................................................................................................................... 42
About Drummond Group .............................................................................................................. 44
## Test Plan Amendments

<table>
<thead>
<tr>
<th>Date of Change</th>
<th>Document Version</th>
<th>Reason for Change</th>
<th>Summary of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/20/07</td>
<td>3.0</td>
<td>Approved by TEG</td>
<td></td>
</tr>
<tr>
<td>10/04/07</td>
<td>3.0.Errata A</td>
<td>Clarification and minor correction</td>
<td>Clarified MNI-Term step in TC A; corrected misuse of HTTP Redirect binding for Response message to POST binding in several test cases</td>
</tr>
<tr>
<td>10/10/07</td>
<td>3.0.Errata B</td>
<td>Clarification and minor correction</td>
<td>Called out general requirements for all test cases; Made use of SOAP consistent throughout Test Case B; inserted line numbers into layout</td>
</tr>
<tr>
<td>10/16/07</td>
<td>3.0 Errata C</td>
<td>Clarified Test Case D</td>
<td>Clarified which SPs (A or B) and IDPs (A or B) is acting in each test step</td>
</tr>
<tr>
<td>10/18/07</td>
<td>3.0 Errata D.</td>
<td>Clarified Test Case E</td>
<td>Changed to using HTTP Redirect binding to be consistent across test case</td>
</tr>
<tr>
<td>10/30/07</td>
<td>3.0 Errata E</td>
<td>Clarified Test Case G</td>
<td>Changed G.4 to use SOAP binding to be consistent with previous step of using Artifact Resolution through SOAP</td>
</tr>
<tr>
<td>10/31/07</td>
<td>3.0 Errata F</td>
<td>Clarified Test Case F</td>
<td>Clarified actions of User A in different browser sessions</td>
</tr>
<tr>
<td>11/01/07</td>
<td>3.0 Errata G</td>
<td>Clarified Test Case G</td>
<td>Removed the must of logging out user when federation is terminated.</td>
</tr>
<tr>
<td>11/05/07</td>
<td>3.0 Errata H</td>
<td>Clarified Test Case B, D, J, K and M</td>
<td>Corrected IDP used in D.3 and added SLO step in D.4. Clarified use of Redirect binding in TC B. Corrected TC M to show proper steps in SAML URI binding and request assertion by ID. Removed redundant request assertion by ID steps in TC J and K</td>
</tr>
<tr>
<td>11/13/07</td>
<td>3.0 Errata I</td>
<td>Clarified TC N</td>
<td>For TC H, changed steps 5 &amp; 6 to make clear they are for backing system out of affiliations; For TC H, changed POST binding to Redirect; In, N.4 and N.5, changed “message payload” to “assertion to be more clear</td>
</tr>
<tr>
<td>11/27/07</td>
<td>3.0 Errata J</td>
<td>Clarified TC A</td>
<td>Made the MNI binding Redirect in TS 6</td>
</tr>
</tbody>
</table>
Introduction

Overview of Test Plan
This document is the Liberty SAML 2.0 Test Criteria Test Plan which contains the scope of the technical requirements for Liberty certification of SAML 2.0. This document is intended to be publicly viewable through the Liberty Alliance website as well as prospective test participants.

The contents of this document include the test cases for Liberty SAML 2.0 certification as well as additional technical information relevant to testing. The test cases include different test steps which as a whole cover the requirements of the SAML profiles [SAMLProf] and SAML conformance modes [SAMLConf].

Another document, Liberty SAML 2.0 Process Test Plan, contains the detailed testing process and test administration requirements for the SAML 2.0 certification test. The Liberty SAML 2.0 Process Test Plan is available only to registered test participants. While the Process Test Plan is used in completing a certification event, it is not needed to understand the technical expectation for completing SAML 2.0 certification.

Test Plan History
This test plan replaces SAML 2.0 Interoperability Testing Procedure (vs. 2.0) test plan [SAMLTP2]. The majority of content is copied directly from [SAMLTP2] test plan, but main changes from the previous version are the reconstructing of the test steps in to specific test cases and the addition test step details.

- SAML 2.0 Interoperability Testing Procedure, vs. 2.0 (07/07/2006)
- SAML 2.0 Interoperability Testing Procedure, vs. 1.0 (2005)

SAML Conformance Modes
This test plan document contains test cases which cover the nine operational conformance modes of SAML 2.0 and the specific features that are required or optional for each mode. The details of each mode are provided in [SAMLConf], and the conformance modes a listed here:

- IdP – Identity Provider
- IdP Lite – Identity Provider Lite
- SP – Service Provider
- SP Lite – Service Provider Lite
- ECP – Enhanced Client/Proxy
- IdP Extended – Identify Provider Extended
- SP Extended – Service Provider Extended
- SAML Attribute Authority
• SAML Authorization Decision Authority
• SAML Authentication Authority
• SAML Requester

Each conformance mode requires different test cases, but some test cases cover multiple
conformance modes. The required test cases for each conformance mode are noted in the Test Case
section of this document.

Certification in conformance modes IdP Extended and SP Extended can only be given if a
participant has met the certification requirements of one of the standard SP or IdP modes.

Because significant features in some of these modes are Optional the Liberty Interoperability Testing
Program has created an additional designation “Complete” to recognize and differentiate
implementations that demonstrate interoperability of all optional features for a particular mode. The
list of “Complete” interoperability designations is:
• SP Complete
• SAML Requester Complete

GSA Profile

The GSA Profile test case is an optional test case. It follows the SAML 2.0 requirements for the
General Service Administration (GSA) of the US Government. The technical requirements for this
test case come from the GSA SAML Profile in [GSAInterface], [GSAAdoptSchm] and
[GSATechAppr]. These documents should be consulted for further explanation of the GSA
requirements.

POST Binding

Although the POST binding is not included in the SAML SCR, it is permitted with the SAML
specification and has some user deployment. POST Binding is an optional Liberty designation
conformance mode. It involves use of POST binding for AuthnRequest, Name ID Management and
SLO. Certification in the POST Binding mode is done through successfully completing this test case.
Technical Requirements

Metadata
There are no normative requirements in [SAMLConf] regarding the content or processing of metadata as described in [SAMLMeta]. However, for purposes of this certification event, implementations are required to:
- Furnish correct metadata, and
- Process metadata furnished by other testing partners
While metadata is not specified for SAML Attribute Requesters, interoperability with SAML Authorities is very difficult without it, and for this certification event it is required that SAML Attribute Requesters provide metadata as described in the draft metadata extension specification [SAMLMetaExt]. It is not necessary or meaningful for an ECP to produce or consume metadata.

IdP Authentication
SAML does not normatively specify any requirements for user authentication at IdP for Web SSO. In fact, user authentication is explicitly described as “out of scope” [SAMLProf]. However, for purposes of interoperability testing, it is required that IdP implementations offer at least one of these authentication methods:
1. HTTP Basic Auth.
2. HTTP Form Post
3. HTTP Get
Similarly, it is required that user agents, particularly ECP implementations, be able to authenticate using at least one of these methods.

Trivial Processing
Several features specified by SAML (e.g., IdP Proxy) can be implemented such that any request simply returns an error response. While this trivial behavior is, strictly speaking, in conformance with the specifications, it is not meaningful in the context of interoperability testing. Except where explicitly indicated (e.g., for certain Name Identifier formats) all testing steps will require non-trivial responses in order to be deemed successful.

Authentication Contexts
Some of the SAML Modes rely on a well-defined ordering of authentication contexts. The SAML specifications do not normatively specify an ordering [SAMLAuthnCxt] and leave the comparison decisions up to the implementation [SAMLCore]. However, for purposes of testing we will arbitrarily define an ordering of authentication contexts to be used in the tests. This arbitrary listing of authentication class URIs, in order of increasing strength, is:
1. any defined authentication context not listed below
2. urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession
4. urn:oasis:names:tc:SAML:2.0:ac:classes:Password

This ordering should be observed by all implementations testing SAML modes where authentication contexts must be compared. The overall concept of the testing of the Authentication Authority is to create several different assertions using different authentication contexts. Then these are queried using the query terms (“exact”, “better”, “maximum”, “minimum”) and a reference authentication context.

NOTE: Complete implementation of these authentication contexts is not required. These authentication context URIs should simply be asserted in requests and responses to demonstrate interoperability of authentication context processing rules.

### Name Identifier Formats

The following Name Identifier Formats are defined by [SAMLCore]:

1. Unspecified
2. Email
3. X.509 Subject
4. Windows
5. Kerberos
6. Entity
7. Persistent
8. Transient

Every implementation is required to accept messages containing any of these formats, but [SAMLCore] only requires that the last two be processed.

### XML Signatures

The [SAMLConf] does not specifically indicate where XML Signatures are required, but the underlying specifications in [SAMLProf] make signing required for certain profiles. Specifically, these are:

1. Web SSO: The assertion element(s) in the <Response> MUST be signed for the HTTP POST binding.
2. ECP Profile: The assertion element(s) in the <Response> issued by the IdP MUST be signed.
3. Single Logout: The <LogoutRequest> and <LogoutResponse> MUST be signed for the HTTP redirect binding.
4. Name Identifier Management: The <ManageNameIDRequest> and <ManageNameIDResponse> MUST be signed for the HTTP redirect binding.
SP and IdP implementations may indicate via metadata a desire for requests or responses to be signed for other bindings than those indicated above. However, such stipulations in metadata are not binding and adherence is not required.

XML Encryption

[SAMLConf] stipulates several different encryption algorithms and key transport mechanisms that MUST be implemented. However, these testing procedures do not require demonstration of support for all these combinations and instead rely on successful interoperability as a measure of conformance. Implementations should take care to ensure that elements to be encrypted include any XML namespace prefix declarations so that, when decrypted, the element will remain valid independent of context. One method for achieving this is described in [ExcXMLCan], but other approaches will work.

Note that while the <ds:KeyInfo> and <xenc:EncryptedKey> elements are not required in the SAML specifications or related schemas, these elements MUST be included in messages for interoperability testing. There is no normative mechanism for exchanging these keys out-of-band. The precise location of these elements in the message is underspecified; the most common practice among interoperable SAML implementations is that in each encrypted element there be one <xenc:EncryptedKey> element in parallel with the <xenc:EncryptedData>, and that this <xenc:EncryptedKey> be inferred as the relevant key information for decryption without relying on any references within the subelements. An erratum has been created to clarify this; see PE43 in [SAMLErrata]. For this certification event, this most common practice stated above SHOULD be done.

Finally, encryption coupled with deflation and URL encoding may create URLs that exceed the maximum length supported by some browsers. Consequently, encryption is contraindicated for the MNI HTTP-Redirect testing steps.

Attribute Profiles

[SAMLConf] makes no normative statements about which Attribute Profiles in [SAMLProf] are required to be supported by SAML Attribute Authority or a SAML Requestor. These are the profiles described in [SAMLProf] except for X.500/LDAP which is described in [SAMLLDAP]:

1. Basic
2. X.500/LDAP
3. UUID
4. DCE PAC
5. XACML

Of these, this document only describes testing procedures for the Basic profile, and does not describe any testing procedures regarding the other profiles.
Test Cases

Overview of Test Case Description

Each test case is setup with the first part listing an overview of the test steps in the test case. The second part describes the details of the individual test steps to carry out the test case. The test step overview lists the sequence of test steps along with a general description of the message or action or configuration setting required. The test step details provide more information on the expected test steps.

Test Cases Associated with Conformance Modes

In order to achieve certification in one or more of the Liberty SAML Conformance Modes, the associated test cases must be completed with all test participants with aligning modes. For example, a product testing for an IdP conformance mode must complete Test Cases A, B, C, E, F, G, H and I against all products testing for a SP and SP Lite conformance mode. The specific pairing among participants will be given at the beginning of the certification event. A conformance mode may not require the completing of all the test steps in the associated test cases. The individual test cases provide details of test steps that may or must be omitted depending on the conformance mode.

<table>
<thead>
<tr>
<th>Conformance Mode</th>
<th>Test Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdP</td>
<td>A, B, E, F, G, H, I, N</td>
</tr>
<tr>
<td>IdP Extended</td>
<td>D</td>
</tr>
<tr>
<td>IdP Lite</td>
<td>A, B, E, F, G, H, I</td>
</tr>
<tr>
<td>SP</td>
<td>A, B, C, E, F, G, H, I, N</td>
</tr>
<tr>
<td>SP Extended</td>
<td>D</td>
</tr>
<tr>
<td>SP Lite</td>
<td>A, B, C, E, F, G, H, I, N</td>
</tr>
<tr>
<td>ECP</td>
<td>I</td>
</tr>
<tr>
<td>SAML Attribute Authority</td>
<td>K</td>
</tr>
<tr>
<td>SAML Authorization Decision Authority</td>
<td>L</td>
</tr>
<tr>
<td>SAML Authentication Authority</td>
<td>J</td>
</tr>
<tr>
<td>SAML Requester</td>
<td>M</td>
</tr>
</tbody>
</table>

Copyright © Drummond Group Inc. 2007
All rights, domestic and international, reserved. This document, whole or in part, may not be distributed to any entity except with written permission from Drummond Group Inc.
General Test Case Requirements

For all test cases, the following requirements are to be followed unless a test case specifically states otherwise:

- SAML AuthnRequest MUST be signed.
- For POST bindings, the assertion MUST be signed.
- For POST bindings, the entire response message MAY be signed, but if signed, the receiving partner MUST validate the signature.
- Encryption of NameIDs and Assertions MUST be enabled.

Test Case A – Redirect Binding

Preconditions: Metadata exchanged and loaded

Conformance Modes: IdP, SP, IdP Lite, SP Lite

NOTE – IdP Lite and SP Lite actors are to ignore Name ID Management steps

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action/Message/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Encryption Enabled</td>
</tr>
<tr>
<td>2</td>
<td>Web SSO HTTP redirect / Persistent / Federate</td>
</tr>
<tr>
<td>3</td>
<td>MNI IdP-Initiated / HTTP redirect (signed)</td>
</tr>
<tr>
<td>4</td>
<td>SLO SP-Initiated / HTTP redirect (signed)</td>
</tr>
<tr>
<td>5</td>
<td>Web SSO HTTP redirect / Not Federated</td>
</tr>
<tr>
<td>6</td>
<td>SLO IdP-initiated / HTTP redirect (signed)</td>
</tr>
<tr>
<td></td>
<td>Destroy Federation and NameIds</td>
</tr>
<tr>
<td>7</td>
<td>Web SSO HTTP redirect / Federate</td>
</tr>
<tr>
<td>8</td>
<td>MNI SP-Initiated / HTTP redirect (signed)</td>
</tr>
<tr>
<td>9</td>
<td>SLO SP-Initiated / HTTP redirect (signed)</td>
</tr>
<tr>
<td>10</td>
<td>Web SSO HTTP redirect</td>
</tr>
<tr>
<td>11</td>
<td>SLO IdP-Initiated / HTTP redirect (signed)</td>
</tr>
<tr>
<td>12</td>
<td>Encryption Disabled</td>
</tr>
</tbody>
</table>

Test Steps Details

1. User is logged into SP. IdP enables encryption of NameId and Assertion.
   
   IdP CONFIRM: IdP has enabled encryption for NameId and Assertion.

2. User/SP does Single Sign-On with Persistent Name Identifier and with Federate where AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is
through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
   IdP CONFIRM: User has been federated
   SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

3. IdP sends signed ManageNameIdRequest message to the SP using HTTP Redirect binding. SP returns signed ManageNameIdResponse message using HTTP Redirect binding. SP Lite and IdP Lite modes omit this step.

   SP CONFIRM: Receives signed ManageNameIdRequest on HTTP Redirect binding.
   IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

4. SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message.

   IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
   SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

5. User/SP does Single Sign-On with Federation already done where AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
   SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

6. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP Redirect binding. SP returns a signed LogoutResponse message. IdP sends signed ManageNameIdRequest message with the Terminate element to the SP using HTTP Redirect binding. Federation for User is terminated. SP returns signed ManageNameIdResponse message using HTTP Redirect binding. User logs out of session.

   IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
   SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
   SP CONFIRM: Receives signed ManageNameIdRequest with Terminate flag on HTTP Redirect binding.
   SP CONFIRM: User federation is terminated.
   SP CONFIRM: User logs out of session.
   IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.
   IdP CONFIRM: User federation is terminated.
   IdP CONFIRM: User logs out of session.

7. User/SP does Single Sign-On with Federate where AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.
IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

8. SP sends signed ManageNameIdRequest message to the IdP using HTTP Redirect binding. IdP returns signed ManageNameIdResponse message using HTTP Redirect binding. SP Lite and IdP Lite modes omit this step.

   IdP CONFIRM: Receives signed ManageNameIdRequest on HTTP Redirect binding.
   SP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

9. SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message.

   IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
   SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

10. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
   SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

11. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP Redirect binding. SP returns a signed LogoutResponse message.

   SP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
   IdP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

12. IdP disables encryption of NameIDs and Assertions. Steps 2-11 are repeated.

   IDP CONFIRM: IdP has disabled encryption for NameId and Assertion.
   IDP/SP CONFIRM: Steps 2-11 are successfully executed.
Test Case B – SOAP Binding

Preconditions: Metadata exchanged and loaded

Conformance Modes: IdP, SP, IdP Lite, SP Lite

NOTE – IdP Lite and SP Lite actors are to ignore Name ID Management steps

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action/Message/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Encrypted IDs and Assertions</td>
</tr>
<tr>
<td>2</td>
<td>Web SSO Artifact / Persistent / Federate / Artifact Resolution SOAP</td>
</tr>
<tr>
<td>3</td>
<td>MNI IdP-Initiated / SOAP (SP may use HTTP Redirect)</td>
</tr>
<tr>
<td>4</td>
<td>SLO SP-Initiated / SOAP (SP Lite / IdP Lite may use HTTP Redirect)</td>
</tr>
<tr>
<td>5</td>
<td>Web SSO Artifact / Persistent / Not Federate / Artifact Resolution SOAP</td>
</tr>
<tr>
<td>6</td>
<td>SLO IdP-initiated / SOAP (SP Lite / IdP Lite may use HTTP Redirect)</td>
</tr>
<tr>
<td>7</td>
<td>Web SSO Artifact / Artifact Resolution SOAP</td>
</tr>
<tr>
<td>8</td>
<td>MNI SP-Initiated / SOAP (SP may use HTTP Redirect)</td>
</tr>
<tr>
<td>9</td>
<td>SLO IdP-Initiated / SOAP (SP Lite / IdP Lite may use HTTP Redirect)</td>
</tr>
<tr>
<td>10</td>
<td>Web SSO Artifact / Artifact Resolution SOAP</td>
</tr>
<tr>
<td>11</td>
<td>SLO IdP-Initiated / HTTP redirect (signed)</td>
</tr>
</tbody>
</table>

Test Steps Details

1. User is logged into SP. IdP enables encryption of NameId and Assertions.

   IdP CONFIRM: IdP has enabled encryption.

2. User/SP does Single Sign-On with Persistent Name Identifier and with Federate where AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is sent through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through Artifact binding. The IdP and SP resolve the artifact via a SOAP binding.

   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

   IdP CONFIRM: Artifact resolution is properly done.

   IdP CONFIRM: User has been federated

   SP CONFIRM: IdP returns signed SAML Response through HTTP Artifact binding.

   SP CONFIRM: Artifact resolution is properly done.
3. IdP sends signed ManageNameIdRequest message to the SP using SOAP binding (SP modes may use HTTP Redirect binding). SP returns signed ManageNameIdResponse message using SOAP binding. SP Lite and IdP Lite modes omit this step.

   SP CONFIRM: Receives signed ManageNameIdRequest on SOAP binding (or possibly HTTP Redirect binding for SP modes).

   IdP CONFIRM: Receives signed ManageNameIdResponse on SOAP binding (or possibly HTTP Redirect binding for SP modes).

4. SP sends a signed LogoutRequest message to IdP using SOAP binding (SP Lite and IdP Lite modes may use HTTP Redirect binding). IdP logs out User session. IdP returns a signed LogoutResponse message.

   IdP CONFIRM: Receives signed LogoutRequest on SOAP binding (or possibly HTTP Redirect binding for SP Lite and IdP Lite modes).

   SP CONFIRM: Receives signed LogoutResponse on SOAP binding (or possibly HTTP Redirect binding for SP Lite and IdP Lite modes).

5. User/SP does Single Sign-On with Federation already done where AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is sent through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through Artifact binding. The IdP and SP resolve the artifact via a SOAP binding.

   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect Binding.

   IdP CONFIRM: Artifact resolution is properly done.

   SP CONFIRM: IdP returns signed SAML Response through HTTP Artifact binding.

   SP CONFIRM: Artifact resolution is properly done.

6. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using SOAP binding (SP Lite and IdP Lite modes may use HTTP Redirect binding). SP returns a signed LogoutResponse message.

   IdP CONFIRM: Receives signed LogoutRequest on SOAP binding (or possibly HTTP Redirect binding for SP Lite and IdP Lite modes).

   SP CONFIRM: Receives signed LogoutResponse on SOAP binding (or possibly HTTP Redirect binding for SP Lite and IdP Lite modes).

7. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through Artifact binding. The IdP and SP resolve the artifact via SOAP binding.

   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

   IdP CONFIRM: Artifact resolution is properly done.

   IdP CONFIRM: User has been federated

   SP CONFIRM: IdP returns signed SAML Response through HTTP Artifact binding.

   SP CONFIRM: Artifact resolution is properly done.
8. SP sends signed ManageNameIdRequest message to the IdP using SOAP binding (SP modes may use HTTP Redirect binding). IdP returns signed ManageNameIdResponse message using SOAP binding. SP Lite and IdP Lite modes omit this step.

   - **IdP CONFIRM**: Receives signed ManageNameIdRequest on SOAP binding (or possibly HTTP Redirect binding for SP modes).
   - **SP CONFIRM**: Receives signed ManageNameIdResponse on SOAP binding (or possibly HTTP Redirect binding for SP modes).

9. IdP logs out User session. IdP sends a signed LogoutRequest message to IdP using SOAP binding (SP Lite and IdP Lite modes may use HTTP Redirect binding). SP returns a signed LogoutResponse message.

   - **SP CONFIRM**: Receives signed LogoutRequest on SOAP binding (or possibly HTTP Redirect binding for SP Lite and IdP Lite modes).
   - **IdP CONFIRM**: Receives signed LogoutResponse on SOAP binding (or possibly HTTP Redirect binding for SP Lite and IdP Lite modes).

10. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through Artifact binding. The IdP and SP resolve the artifact via a SOAP binding.

   - **IdP CONFIRM**: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
   - **IdP CONFIRM**: Artifact resolution is properly done.
   - **SP CONFIRM**: IdP returns signed SAML Response through HTTP Artifact binding.
   - **SP CONFIRM**: Artifact resolution is properly done.

11. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using SOAP binding (SP Lite and IdP Lite modes may use HTTP Redirect binding). SP returns a signed LogoutResponse message.

   - **IdP CONFIRM**: Receives signed LogoutRequest on SOAP binding (or possibly HTTP Redirect binding for SP Lite and IdP Lite modes).
   - **SP CONFIRM**: Receives signed LogoutResponse on SOAP binding (or possibly HTTP Redirect binding for SP Lite and IdP Lite modes).
Test Case C – POST Binding

Preconditions: Metadata exchanged and loaded
Conformance Modes: POST Binding

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action/Message/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Encrypted Enabled</td>
</tr>
<tr>
<td>2</td>
<td>Web SSO / POST (signed) / Persistent / Federate</td>
</tr>
<tr>
<td>3</td>
<td>MNI IdP-Initiated / POST (signed)</td>
</tr>
<tr>
<td>4</td>
<td>SLO SP-Initiated / POST (signed)</td>
</tr>
<tr>
<td>5</td>
<td>Web SSO POST (signed) / Not Federated</td>
</tr>
<tr>
<td>6</td>
<td>SLO IdP-initiated / POST (signed)</td>
</tr>
<tr>
<td>7</td>
<td>Web SSO POST (signed)</td>
</tr>
<tr>
<td>8</td>
<td>MNI IdP-initiated / POST / Terminate</td>
</tr>
<tr>
<td>9</td>
<td>Web SSO POST (signed)</td>
</tr>
<tr>
<td>10</td>
<td>MNI SP-Initiated / POST (signed)</td>
</tr>
<tr>
<td>11</td>
<td>SLO SP-Initiated / POST (signed)</td>
</tr>
<tr>
<td>12</td>
<td>Web SSO POST (signed)</td>
</tr>
<tr>
<td>13</td>
<td>SLO IdP-Initiated / POST (signed)</td>
</tr>
</tbody>
</table>

Test Steps Details

1. User is logged into SP. IdP enables encryption of NameId and Assertion.
   
   IdP CONFIRM: IdP has enabled encryption.

2. User/SP does Single Sign-On with Persistent Name Identifier and with Federate where
   AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is
   through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML
   Response message through HTTP POST binding.
   
   IdP CONFIRM: SP successfully communicated SAML Authentication Request through
   HTTP POST binding.
   
   IdP CONFIRM: User has been federated
   
   SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

3. IdP sends signed ManageNameIdRequest message to the SP using HTTP POST binding. SP
   returns signed ManageNameIdResponse message using HTTP POST binding.
   
   SP CONFIRM: Receives signed ManageNameIdRequest on HTTP POST binding.
   
   IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.

4. SP sends a signed LogoutRequest message to IdP using HTTP POST binding. IdP logs out User
   session. IdP returns a signed LogoutResponse message.
IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.

SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.

5. User/SP does Single Sign-On with Federation already done where AllowCreate is set to FALSE.

SP confirmation to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.
   SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

6. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP POST binding. SP returns a signed LogoutResponse message.

   IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.
   SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.

7. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.
   SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

8. IdP sends signed ManageNameIdRequest message with the Terminate element to the SP using HTTP POST binding. User session is terminated. SP returns signed ManageNameIdResponse message using HTTP POST binding.

   SP CONFIRM: Receives signed ManageNameIdRequest with Terminate flag on HTTP POST binding.
   SP CONFIRM: User session is terminated.
   IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.
   IdP CONFIRM: User session is terminated.

9. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.
   SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

10. SP sends signed ManageNameIdRequest message to the IdP using HTTP POST binding. IdP returns signed ManageNameIdResponse message using HTTP POST binding.

   IdP CONFIRM: Receives signed ManageNameIdRequest on HTTP POST binding.
   SP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.
11. SP sends a signed LogoutRequest message to IdP using HTTP POST binding. IdP logs out User session. IdP returns a signed LogoutResponse message.
   
   IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.
   
   SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.

12. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.
   
   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.
   
   SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

13. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP POST binding. SP returns a signed LogoutResponse message.
   
   SP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.
   
   IdP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.
Test Case D – Extended SAML Modes

Preconditions: Metadata exchanged and loaded
Conformance Modes: IdP Extended, SP Extended

Background on IdP Proxy
The IdP Proxy feature requires two IdP implementations and one SP implementation. If we have teams A and B, the following diagram depicts the roles of the test participants, assuming that IdP_A and SP_B are the implementations under test:

Background on Name Identifier Mapping Feature
The name identifier mapping feature requires that an IdP provide an indirect reference for a principal at SP_A in response to a request from SP_B. Assuming again that teams A and B are testing IdP_A and SP_B, it is necessary for the principal to federate her identity at both SP_B and SP_A with IdP_A. This can be depicted as follows:

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action/Message/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Encryption Enabled</td>
</tr>
<tr>
<td>2</td>
<td>ProxyCount=0 (proxy disallowed)</td>
</tr>
<tr>
<td>3</td>
<td>Web SSO HTTP Redirect (signed) to IdP_A</td>
</tr>
<tr>
<td>4</td>
<td>SLO SP-initiated / HTTP Redirect</td>
</tr>
<tr>
<td>5</td>
<td>ProxyCount missing (proxy allowed)</td>
</tr>
<tr>
<td>6</td>
<td>Web SSO / HTTP Redirect (signed) to IdP_A</td>
</tr>
<tr>
<td>7</td>
<td>SLO SP-initiated / HTTP Redirect</td>
</tr>
</tbody>
</table>
8. ProxyCount=1 (proxy allowed)

9. Web SSO / HTTP Redirect (signed)

10. SLO SP-initiated / HTTP Redirect

11. Web SSO HTTP Redirect (signed) / Persistent

12. SLO IdP-initiated / HTTP Redirect

13. NameIDMappingRequest / NameIDMappingResponse

Test Steps Details

1. IdP_A and IdP_B enables encryption of NameId and Assertion.
   - IdP_A CONFIRM: IdP_A has enabled encryption.
   - IdP_B CONFIRM: IdP_B has enabled encryption.

2. SP sets ProxyCount=0 where proxy is disallowed.
   - SP CONFIRM: SP has disallowed proxy.

3. User/SP does Single Sign-On. SP communication to the IdP_A for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User and IdP_A returns a signed SAML Response message through HTTP POST binding.
   - IdP_A CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
   - IdP_A CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
   - SP CONFIRM: IdP_A returns signed SAML Response through HTTP POST binding.

4. SP sends a signed LogoutRequest message to IdP_A using HTTP Redirect binding. IdP_A logs out User session. IdP_A returns a signed LogoutResponse message.
   - IdP_A CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
   - SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

5. SP removes ProxyCount where proxy is allowed.
   - SP CONFIRM: SP has allowed proxy.

6. User/SP does Single Sign-On. SP communication to the IdP_A for the SAML Authentication Request is through HTTP Redirect binding. IdP_A proxies the Authentication Request to IdP_B. IdP_B provides assertion of User to IdP_A in a SAML Response message through HTTP POST binding and IdP_A returns a signed SAML Response message through HTTP POST binding.
   - IdP_A CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
   - IdP_B CONFIRM: IdP_A proxies Authentication Request.
   - SP CONFIRM: IdP_A returns signed SAML Response through HTTP POST binding.

7. SP sends a signed LogoutRequest message to IdP_A using HTTP Redirect binding. IdP_A logs out User session. IdP_A returns a signed LogoutResponse message.
   - IdP_A CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
   - SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
8. SP sets ProxyCount=1 where proxy is allowed.

   SP CONFIRM: SP has allowed 1 proxy.

9. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication
    Request is through HTTP Redirect binding. IdP_A proxies the Authentication Request to IdP_B. IdP_B
    provides assertion of User to IdP_A in a SAML Response message through HTTP POST binding and
    IdP_A returns a signed SAML Response message through HTTP POST binding.

    IdP_A CONFIRM: SP successfully communicated SAML Authentication Request through
    HTTP Redirect binding.

    IdP_B CONFIRM: IdP_A proxies Authentication Request.

    SP CONFIRM: IdP_A returns signed SAML Response through HTTP POST binding.

10. SP sends a signed LogoutRequest message to IdP_A using HTTP Redirect binding. IdP_A logs out
    User session. IdP_A returns a signed LogoutResponse message.

    IdP_A CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

    SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

11. User/SP_A does Single Sign-On with Persistent Name Identifier. SPA communication to the IdP
    for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of
    User and IdP returns a signed SAML Response message through HTTP POST binding.

    IdP CONFIRM: SPA successfully communicated SAML Authentication Request through
    HTTP POST binding.

    IdP CONFIRM: User has been federated

    SP_A CONFIRM: IdP returns signed SAML Response through HTTP Redirect binding.

12. IdP logs out User session. IdP sends a signed LogoutRequest message to SP_A using HTTP
    Redirect binding. SP_A returns a signed LogoutResponse message.

    SP_A CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

    IdP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

13. SP_B sends NameIdMapping Request message to the IdP requesting an alternative name identifier
    for User. IdP maps the request to the User identity from SP_A. IdP returns signed
    NameIdMappingResponse message using HTTP POST binding.

    IdP CONFIRM: Receives NameIdMapping Request.

    SP CONFIRM: Receives NameIdMappingResponse Response.
Test Case E – IDP Introduction

Preconditions: Metadata exchanged and loaded

Conformance Modes: IdP, SP, IdP Lite, SP Lite

NOTE – IdP Lite and SP Lite actors are to ignore Name ID Management steps

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enables Encryption / Clear Cookies</td>
</tr>
<tr>
<td>2</td>
<td>IdP Login / Federate / Set Cookie</td>
</tr>
<tr>
<td>3</td>
<td>SSO at SP using common domain cookie</td>
</tr>
<tr>
<td>4</td>
<td>MNI Destroy Federation / IdP-Initiated / HTTP Redirect</td>
</tr>
</tbody>
</table>

Test Step Detail

1. IdP enables encryption of NameId and Assertion. Cookies are cleared from User Browser
   
   **IdP CONFIRM:** IdP has enabled encryption.

2. User logs in at IdP with Federation. Cookie is set.
   
   **IdP CONFIRM:** User logged in cookie is set.

3. User/SP does Single Sign-On using a common domain cookie. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.
   
   **IdP CONFIRM:** SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
   
   **SP CONFIRM:** IdP returns signed SAML Response through HTTP POST binding.

4. IdP sends signed ManageNameIdRequest message with the Terminate element to the SP using HTTP Redirect binding. User session is terminated. SP returns signed ManageNameIdResponse message using HTTP Redirect binding.
   
   **SP CONFIRM:** Receives signed ManageNameIdRequest with Terminate flag on HTTP Redirect binding.
   
   **SP CONFIRM:** User session is terminated.
   
   **IdP CONFIRM:** Receives signed ManageNameIdResponse on HTTP Redirect binding.
   
   **IdP CONFIRM:** User session is terminated.
### Test Case F – Single Session Logout

**Preconditions:** Metadata exchanged and loaded

**Conformance Modes:** IdP, SP, IdP Lite, SP Lite

NOTE – IdP Lite and SP Lite actors are to ignore Name ID Management steps

#### Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action/Message/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Web SSO (Browser A) / Federate / HTTP Redirect</td>
</tr>
<tr>
<td>2</td>
<td>Web SSO (Browser B) HTTP Redirect</td>
</tr>
</tbody>
</table>
| 3     | SLO (Browser A) SP-Initiated / HTTP redirect (signed)  
Browser B session remains active |
| 4     | Web SSO (Browser A) HTTP Redirect |
| 5     | SLO (Browser A) IdP-Initiated / HTTP redirect (signed)  
Browser B session remains active |
| 6     | MNI SP-initiated (Browser B) / Redirect (signed) / Terminate |

#### Test Steps

1. User A/Browser A does Single Sign-On with Federate where AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User A and IdP returns a signed SAML Response message through HTTP POST binding.

   **IdP CONFIRM:** SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

   **IdP CONFIRM:** User A has been federated.

   **SP CONFIRM:** IdP returns signed SAML Response through HTTP POST binding.

2. User A/Browser B does Single Sign-On. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User B and IdP returns a signed SAML Response message through HTTP POST binding.

   **IdP CONFIRM:** SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

   **IdP CONFIRM:** User B has been logged in through Browser A (Session A).

   **SP CONFIRM:** IdP returns signed SAML Response through HTTP POST binding.

3. User A/Browser A logs off of SP. SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User A in Browser A (Session A). User A remains logged in through Browser B (Session B). IdP returns a signed LogoutResponse message.

   **IdP CONFIRM:** Receives signed LogoutRequest on HTTP Redirect binding.

   **IdP CONFIRM:** User A logs out in Browser A (Session A).

   **IdP CONFIRM:** User A is logged in through Browser B (Session B).
4. User A/Browser A does Single Sign-On. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User A and IdP returns a signed SAML Response message through HTTP POST binding.

   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

   IdP CONFIRM: User A has been logged in through Browser A (Session A).

   SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

5. User A/Browser A logs off of IdP. IdP sends a signed LogoutRequest message to SP using HTTP Redirect binding. SP logs out User A in Browser A (Session A). User A remains logged in through Browser B (Session B). SP returns a signed LogoutResponse message.

   SP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

   SP CONFIRM: User A logs out in Browser A (Session A).

   SP CONFIRM: User A is logged in through Browser B (Session B).

   IdP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

   IdP CONFIRM: User A logs out in Browser A (Session A).

   IdP CONFIRM: User A is logged in through Browser B (Session B).

6. SP sends signed ManageNameIdRequest message with the Terminate element to the IdP using HTTP Redirect binding. Federation for User A is terminated. IdP returns signed ManageNameIdResponse message using HTTP Redirect binding. User A logs out of Browser B (Session B).

   IdP CONFIRM: Receives signed ManageNameIdRequest with Terminate flag on HTTP Redirect binding.

   IdP CONFIRM: Federation of User A is terminated.

   IdP CONFIRM: User A is logged out in Browser B (Session B).

   SP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

   SP CONFIRM: Federation of User A is terminated.

   SP CONFIRM: User A is logged out in Browser B (Session B).
Test Case G – Unsolicited <Response>

Preconditions: Metadata exchanged and loaded

Conformance Modes: IdP, SP, IdP Lite, SP Lite

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IdP Unsolicited SSO Response / Transient / HTTP POST (signed)</td>
</tr>
<tr>
<td>2</td>
<td>SLO SP-Initiated / HTTP redirect (signed)</td>
</tr>
<tr>
<td>3</td>
<td>IdP Unsolicited SSO Response / Transient / HTTP artifact / Artifact Resolution (SOAP)</td>
</tr>
<tr>
<td>4</td>
<td>SLO IdP-Initiated / HTTP redirect (signed)</td>
</tr>
</tbody>
</table>

Test Step Detail

1. User/IdP does Single Sign-On. IdP provides assertion of User and Name ID is Transient. IdP sends a signed SAML Response message through HTTP POST binding.
   - IdP CONFIRM: User has been federated
   - SP CONFIRM: IdP sends signed SAML Response through HTTP POST binding.

2. SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message.
   - IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
   - SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

3. User/IdP does Single Sign-On. IdP provides assertion of User and Name ID is Transient. IdP sends a signed SAML Response message through HTTP Artifact using an HTTP Redirect URL. The IdP and SP resolve the artifact via a SOAP binding.
   - IdP CONFIRM: Artifact resolution is properly done.
   - SP CONFIRM: User has been federated
   - IdP CONFIRM: User has been federated
   - SP CONFIRM: IdP sends signed SAML Response through HTTP Artifact.
   - SP CONFIRM: Artifact resolution is properly done.

4. IdP sends a signed LogoutRequest message to SP using SOAP binding. SP logs out User session. SP returns a signed LogoutResponse message.
   - SP CONFIRM: Receives signed LogoutRequest on SOAP binding.
   - IdP CONFIRM: Receives signed LogoutResponse on SOAP binding.
Test Case H – Affiliations

Preconditions: Metadata exchanged and loaded
Conformance Modes: IdP, SP, IdP Lite, SP Lite

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SPNameQualifier=[affiliation id]</td>
</tr>
<tr>
<td>2</td>
<td>Web SSO HTTP Redirect / Persistent / Federate</td>
</tr>
<tr>
<td>3</td>
<td>SLO IdP-initiated / HTTP Redirect (signed)</td>
</tr>
<tr>
<td>4</td>
<td>Web SSO HTTP Redirect / Not Federate</td>
</tr>
<tr>
<td>5</td>
<td>SLO SP-initiated / HTTP Redirect (signed)</td>
</tr>
<tr>
<td>6</td>
<td>SPNameQualifier=[sp provider id]</td>
</tr>
</tbody>
</table>

Test Step Detail

1. SP sets Name Qualifier to Affiliation ID.
   
   **SP CONFIRM:** SPNameQualifier=[affiliation id]

2. User/SP does Single Sign-On with Persistent Name Identifier and with Federate where AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

   **IdP CONFIRM:** SP successfully communicated SAML Authentication Request through HTTP POST binding.
   
   **IdP CONFIRM:** User has been federated

3. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP Redirect binding. SP returns a signed LogoutResponse message.

   **SP CONFIRM:** Receives signed LogoutRequest on HTTP Redirect binding.
   
   **IdP CONFIRM:** Receives signed LogoutResponse on HTTP Redirect binding.

4. User/SP does Single Sign-On with Federation already done where AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

   **IdP CONFIRM:** SP successfully communicated SAML Authentication Request through HTTP POST binding.

5. SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message.

   **IdP CONFIRM:** Receives signed LogoutRequest on HTTP Redirect binding.
SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

6. SP sets Name Qualifier to SP Provider ID.
   SP CONFIRM: SPNameQualifier=[sp provider id]

NOTE: It is recommended that the SP verifies it has changed back the SPNameQualifier through a SSO/SLO test with other participants.
Test Case I – ECP

Preconditions: Metadata exchanged and loaded

Conformance Modes: IdP, SP, IdP Lite, SP Lite, ECP

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Federate (NameIDPolicy, AllowCreate=True)</td>
</tr>
<tr>
<td>2</td>
<td>Enhanced ClientProxy SSO, PAOS</td>
</tr>
<tr>
<td>3</td>
<td>ECP conveys Response to SP</td>
</tr>
</tbody>
</table>

Test Step Detail

1. User attempts to access SP through ECP. Settings are Persistent Name Identifier and with Federate where AllowCreate is set to TRUE to SP.
   - ECP CONFIRM: Connects to SP.
   - SP CONFIRM: ECP connects.

2. SP issues SAML AuthnRequest message to ECP through PAOS binding. ECP sends SAML Request message to IdP through SOAP binding. IdP returns assertion in SAML Response to ECP.
   - IdP CONFIRM: ECP successfully communicated SAML Authentication Request through SOAP binding.
   - SP CONFIRM: IdP returns signed SAML Response through HTTP SOAP binding.

3. ECP conveys Response to SP. SP grants access to User. User closes browser and session ends.
   - SP CONFIRM: Proper Response is received.
   - ECP CONFIRM: User is granted access to SP.
Test Case J – SAML Authentication Authority

Conformance Modes: SAML Authentication Authority

Preconditions: Metadata exchanged and loaded

Note: Section [AuthenticationContexts] within this document describes the strength of the AuthnContext classes used for comparison.

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action/Message/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Web SSO / POST (signed) / Persistent</td>
</tr>
<tr>
<td>2</td>
<td>AC Comparison=&quot;exact&quot; / HTTP Basic Authentication</td>
</tr>
<tr>
<td>3</td>
<td>Authentication Query / SOAP</td>
</tr>
<tr>
<td>4</td>
<td>AC Comparison=&quot;better&quot;</td>
</tr>
<tr>
<td>5</td>
<td>Authentication Query / SOAP</td>
</tr>
<tr>
<td>6</td>
<td>AC Comparison=&quot;minimum&quot;</td>
</tr>
<tr>
<td>7</td>
<td>Authentication Query / SOAP</td>
</tr>
<tr>
<td>8</td>
<td>AC Comparison=&quot;maximum&quot;</td>
</tr>
<tr>
<td>9</td>
<td>Authentication Query / SOAP</td>
</tr>
</tbody>
</table>

Test Step Detail

1. User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

   **IdP CONFIRM:** SP successfully communicated SAML Authentication Request through HTTP POST binding.

   **IdP CONFIRM:** User has been federated

   **SP CONFIRM:** IdP returns signed SAML Response through HTTP POST binding.

2. SAML Requester sets AC comparison to “exact”. SAML Requester/Responder enables HTTP Basic Authentication.

   **SAML Requester CONFIRM:** AC comparison=”exact”.

   **SAML Requester CONFIRM:** HTTP Basic Authentication enabled.

   **SAML Responder CONFIRM:** HTTP Basic Authentication enabled.

3. SAML Requester sends Authentication Query to SAML Responder through SOAP binding.

   SAML Responder returns SAML Response.

   **SAML Responder CONFIRM:** SAML Requester sent Authentication Query.

   **SAML Requester CONFIRM:** SAML Responder returned the SAML Response.

4. SAML Requester sets AC comparison to “better”.

   **SAML Requester CONFIRM:** AC comparison=”better”.
5. SAML Requester sends Authentication Query to SAML Responder through SOAP binding.

SAML Responder returns SAML Response.

SAML Responder CONFIRM: SAML Requester sent Authentication Query.

SAML Requester CONFIRM: SAML Responder returned the SAML Response.

6. SAML Requester sets AC comparison to “minimum”.

SAML Requester CONFIRM: AC comparison=”minimum”.

7. SAML Requester sends Authentication Query to SAML Responder through SOAP binding.

SAML Responder returns SAML Response.

SAML Responder CONFIRM: SAML Requester sent Authentication Query.

SAML Requester CONFIRM: SAML Responder returned the SAML Response.

8. SAML Requester sets AC comparison to “maximum”.

SAML Requester CONFIRM: AC comparison=” maximum”.

9. SAML Requester sends Authentication Query to SAML Responder through SOAP binding.

SAML Responder returns SAML Response.

SAML Responder CONFIRM: SAML Requester sent Authentication Query.

SAML Requester CONFIRM: SAML Responder returned the SAML Response.
Test Case K – SAML Attribute Authority

Conformance Modes: SAML Attribute Authority

Preconditions: Metadata exchanged and loaded

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action/Message/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Web SSO / POST (signed) / Persistent</td>
</tr>
<tr>
<td>2</td>
<td>Attribute Query No Attributes</td>
</tr>
<tr>
<td>3</td>
<td>Attribute Query / SOAP</td>
</tr>
<tr>
<td>4</td>
<td>Attribute Query Attribute Named</td>
</tr>
<tr>
<td>5</td>
<td>Attribute Query / SOAP</td>
</tr>
<tr>
<td>6</td>
<td>Attribute Query Attribute Value</td>
</tr>
<tr>
<td>7</td>
<td>Attribute Query / SOAP</td>
</tr>
<tr>
<td>8</td>
<td>Encrypted Attribute / Attribute Query Attribute Named</td>
</tr>
<tr>
<td>9</td>
<td>Attribute Query / SOAP</td>
</tr>
</tbody>
</table>

Test Step Detail

1. User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

   IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.

   IdP CONFIRM: User has been federated

   SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

2. SAML Responder sets attribute query to no attributes.

   SAML Responder CONFIRM: Attribute Query No Attributes.

3. SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.

   SAML Responder CONFIRM: SAML Requester sent Attribute Query.

   SAML Requester CONFIRM: SAML Responder returned the SAML Response.

4. SAML Responder sets attribute query to attribute named.

   SAML Responder CONFIRM: Attribute Query Attribute Named.

5. SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.

   SAML Responder CONFIRM: SAML Requester sent Attribute Query.

   SAML Requester CONFIRM: SAML Responder returned the SAML Response.
6. SAML Responder sets attribute query to attribute value.
   SAML Responder CONFIRM: Attribute Query Attribute Value.

7. SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML
   Responder returns SAML Response.
   SAML Responder CONFIRM: SAML Requester sent Attribute Query.
   SAML Requester CONFIRM: SAML Responder returned the SAML Response.

8. SAML Responder sets attribute query to attribute named. SAML Responder enables attribute for
   encryption.
   SAML Responder CONFIRM: Attribute Query Attribute Named.
   SAML Responder CONFIRM: Encryption assertion enabled.

9. SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML
   Responder returns SAML Response.
   SAML Responder CONFIRM: SAML Requester sent Attribute Query.
   SAML Requester CONFIRM: SAML Responder returned the SAML Response.
Test Case L – SAML Authorization Decision Authority

Conformance Modes: SAML Authorization Decision Authority

Preconditions: Metadata exchanged and loaded

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action/Message/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Web SSO / POST (signed) / Persistent</td>
</tr>
<tr>
<td>2</td>
<td>HTTP Basic Authentication</td>
</tr>
<tr>
<td>3</td>
<td>AuthzQuery Resource=never (never permitted)</td>
</tr>
<tr>
<td>4</td>
<td>Authorization Decision Query / SOAP</td>
</tr>
<tr>
<td>5</td>
<td>AuthzQuery Resource=maybe (permitted if auth match)</td>
</tr>
<tr>
<td>6</td>
<td>Authorization Decision Query / SOAP</td>
</tr>
<tr>
<td>7</td>
<td>AuthzQuery Resource=always (always permitted)</td>
</tr>
<tr>
<td>8</td>
<td>Authorization Decision Query / SOAP</td>
</tr>
</tbody>
</table>

Test Step Detail

1. User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.
   - IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.
   - IdP CONFIRM: User has been federated
   - SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

2. SAML Requester enables HTTP Basic Authentication.
   - SAML Requester CONFIRM: HTTP Basic Authentication enabled.

3. SAML Responder sets Authorization Query to never permitted which means subject is never authorized for access.
   - SAML Responder CONFIRM: AuthzQuery Resource=never

4. SAML Requester sends Authorization Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.
   - SAML Requester CONFIRM: SAML Responder returned the SAML Response.

5. SAML Responder sets authorization query to maybe permitted if authentication is matched which means subject is authorized if it is a “particular” subject.
   - SAML Responder CONFIRM: AuthzQuery Resource=maybe
6. SAML Requester sends Authorization Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.
   SAML Responder CONFIRM: SAML Requester sent Authorization Query.
   SAML Requester CONFIRM: SAML Responder returned the SAML Response.

7. SAML Responder sets Authorization Query to always permitted which means subject is always authorized.
   SAML Responder CONFIRM: AuthzQuery Resource=always

8. SAML Requester sends Authorization Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.
   SAML Responder CONFIRM: SAML Requester sent Authorization Query.
   SAML Requester CONFIRM: SAML Responder returned the SAML Response.
Test Case M – SAML URI Binding

Conformance Modes: SAML Attribute Authority, SAML Authorization Decision, SAML Authentication Authority, SAML Requester

Preconditions: Metadata exchanged and loaded

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action/Message/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HTTP Basic Authentication</td>
</tr>
<tr>
<td>2</td>
<td>Request for Assertion by Identifier / SOAP</td>
</tr>
<tr>
<td>3</td>
<td>HTTP Basic Authentication</td>
</tr>
<tr>
<td>4</td>
<td>SAML URI Binding</td>
</tr>
</tbody>
</table>

Test Step Detail

1. Requester enables HTTP Basic Authentication
   Requester CONFIRM: HTTP Basic Authentication enabled.

2. Requester sends SAML Request message to Responder in SOAP over HTTP. Request message contains <AssertionIDRequest> and HTTP basic authentication. Assertion ID is assigned at test time. Responder returns identified <Assertion> in SAML Response in SOAP over HTTP.
   Requester and Responder CONFIRM: Use SOAP over HTTP
   Requester and Responder CONFIRM: Use HTTP basic authentication.
   Requester and Responder CONFIRM: Request message contains <AssertionIDRequest> for assigned assertion ID.
   Requester and Responder CONFIRM: Response message contains identified <Assertion>.

3. Requester enables HTTP Basic Authentication
   Requester CONFIRM: HTTP Basic Authentication enabled.

4. Requester sends HTTP GET message to Responder with URI containing assertion ID. Assertion ID assigned at test time. HTTP GET message contains HTTP basic authentication. Responder returns HTTP response with SAML Response containing assigned <Assertion>.
   Requester and Responder CONFIRM: Use HTTP basic authentication.
   Requester and Responder CONFIRM: HTTP GET URI contains assigned assertion ID.
   Requester and Responder CONFIRM: Assigned <Assertion> is returned in response.
Test Case N – Error Testing

Conformance Modes: IdP, SP, SP Lite

Preconditions: Metadata exchanged and loaded

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action/Message/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Artifact Refused</td>
</tr>
<tr>
<td>2</td>
<td>Successful Response Message</td>
</tr>
<tr>
<td>3</td>
<td>Repost of Assertion</td>
</tr>
<tr>
<td>4</td>
<td>Altered data, signature mismatch</td>
</tr>
<tr>
<td>5</td>
<td>Wrongkey used to sign</td>
</tr>
<tr>
<td>6</td>
<td>SubjectConfirmation Recipient != assertion service consumer URL</td>
</tr>
<tr>
<td>7</td>
<td>Unknown SubjectConfirmationMethod</td>
</tr>
<tr>
<td>8</td>
<td>IncorrectAudienceRestriction != requestor</td>
</tr>
<tr>
<td>9</td>
<td>SubjectConfirmation NoOnOrAfter expired</td>
</tr>
<tr>
<td>10</td>
<td>Unknown Condition</td>
</tr>
</tbody>
</table>

Test Step Detail

NOTE – Test Steps 2-9 involve the Liberty Error Test Tool. Metadata for conducting these tests will be exchanged.

2. Test Harness POSTs an unsolicited SAML Response message containing a valid assertion.

   SP CONFIRM: SAML Response was received and assertion accepted.

3. Test Harness re-POSTs the assertion that was successful during the initialization of this test sequence.

   SP CONFIRM: Assertions are not replayed within the validity period of the assertion.

4. The assertion of the SAML Response from Step 2 is altered and sent without re-signing in a HTTP POST from Test Harness.

   SP CONFIRM: SP rejects the message.

5. The assertion of the SAML Response from Step 2 is sent but signed with the wrong signing key in a HTTP POST from Test Harness.

   SP CONFIRM: SP rejects the message.

6. The Test Harness constructs a SAML Response message with an incorrect Recipient attribute. Recipient attribute is in the <SubjectConfirmationData> element.

   SP CONFIRM: SP detects and rejects the message.
7. The Test Harness sends an altered assertion in the SAML Response. A different Method URN is substituted in the assertion’s \(<SubjectConfirmation>\) element other than the required Method of \(urn:oasis:names:tc:SAML:2.0:cm:bearer\).

   \[\text{SP CONFIRM: SP detects and rejects the message.}\]

8. The Test Harness POSTs a SAML Response containing an assertion which does not contain an \(<AudienceRestriction>\) including the SP's unique identifier as an \(<Audience>\).

   \[\text{SP CONFIRM: SP rejects the assertion.}\]

9. The Test Harness sets the NotOnOrAfter attribute to a future value which has passed.

   \[\text{SP CONFIRM: The SP to reject the assertion because of the NotOnOrAfter attribute.}\]

10. The Test Harness includes a \(<Condition>\) extension element in the \(<Conditions>\) element of the assertion which cannot be understood.

   \[\text{SP CONFIRM: The SP rejects the assertion.}\]
Test Case O – GSA Profile

Preconditions: Metadata exchanged and loaded
Conformance Modes: IdP, SP

Test Step Overview

<table>
<thead>
<tr>
<th>Steps</th>
<th>Action/Message/Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IdP Discovery</td>
</tr>
<tr>
<td>2</td>
<td>Web SSO AuthnRequest / HTTP Redirect (signed)</td>
</tr>
<tr>
<td>3</td>
<td>Web SSO AuthnResponse / HTTP POST (signed)</td>
</tr>
<tr>
<td>4</td>
<td>SLO SP-initiated / HTTP Redirect(signed)</td>
</tr>
<tr>
<td>5</td>
<td>IdP Discovery</td>
</tr>
<tr>
<td>6</td>
<td>Web SSO at IdP AuthnResponse / HTTP POST (signed)</td>
</tr>
<tr>
<td>7</td>
<td>SLO SP-initiated / HTTP Redirect(signed)</td>
</tr>
</tbody>
</table>

Pre-Test Setup
1. TLS certificates MUST be trusted by default in commonly used browsers. Certificates and security toolkits MUST follow GSA requirements.
2. Common domain name cookies must be properly created with all IdPs.

Test Steps Details
1. User logs into SP through TLS. User/SP does Single Sign-On. SP uses common domain cookie for IdP Discovery to find IdP.

   REQUIRED: The SP MUST present a tailored list of compatible IdP featuring, at a minimum, compatible IdP(s) in the CDC.

2. SP sends a SAML Authentication Request to the IdP through HTTP Redirect binding. Authentication Request must be signed and delivered over TLS. The following list contains either requirements for the request which must occur, conditional actions which if done must be handled in a specific way or optional conditions which may or may not be done.

   <AuthnRequest>

   REQUIRED: <Issuer> MUST be present, MUST be the identifier of the SP and MUST be a URL within the SP domain.
   CONDITIONAL: ProtocolBinding is optional but if present it MUST be urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST.
   CONDITIONAL: <RequestedAuthnContext> MAY be included in authentication request but if present, the Comparison attribute MUST be set to minimum.
   OPTIONAL: IsPassive MAY be used.
   OPTIONAL: Within <NameIDPolicy>, AllowCreate attribute MAY be present.
   CONDITIONAL: SP MAY set ForceAuthn to true. If ForceAuthn is set to TRUE, IsPassive MUST either be omitted or set to FALSE.
CONDITIONAL: If SP sets ForceAuthn to true, IdP MUST authenticate regardless of user’s authentication session status.

CONDITIONAL: Within <NameIDPolicy>, if Format is present, it MUST use one of the following:
- urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
- urn:oasis:names:tc:SAML:2.0:nameid-format:transient

3. IdP returns the Authentication Response with the assertion to the SP through HTTP POST binding over TLS to the SP’s <Assertion> Consumer Service. The following list contains either requirements for the request which must occur or optional conditions which may be done.

**<Response>**
- REQUIRED: Version attribute MUST be set to “2.0”.
- REQUIRED: Each <Response> MUST contain no more than one <EncryptedAssertion>.
- CONDITIONAL: Consent attribute MAY be used, but if used MUST use one of the following:
  - urn:oasis:names:tc:SAML:2.0:consent:obtained
  - urn:oasis:names:tc:SAML:2.0:consent:prior
  - urn:oasis:names:tc:SAML:2.0:consent:current-explicit
  - urn:oasis:names:tc:SAML:2.0:consent:unspecified

**<Assertion>** element within the <Response>
- REQUIRED: An <Assertion> MUST be returned within <EncryptedAssertion> by the IdP and MUST be signed and encrypted.
- REQUIRED: Version MUST be 2.0.
- REQUIRED: <Issuer> MUST be present, MUST be the identifier of the IdP and MUST be a URL reference within the domain of the IdP.
- REQUIRED: There MUST be exactly one <Subject> per <Assertion>.

**<Subject>** element within the <Assertion>
- REQUIRED: An <Assertion> MUST contain exactly one <Subject> indicating the end user to which <Assertion> pertains.
- REQUIRED: <NameID> MUST contain a Format attribute set either:
  - urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
  - urn:oasis:names:tc:SAML:2.0:nameid-format:transient

**<AuthnStatement>** element within the <Assertion>
- REQUIRED: <AuthnStatement> MUST include the SessionIndex of the end user.
- OPTIONAL: <AuthnStatement> MAY contain SessionNotOnOrAfter but SP is NOT REQUIRED to honor SessionNotOnOrAfter.

**<AttributeStatement>** element within the <Assertion>
The first transmission of an <Assertion> MUST contain exactly one <AttributeStatement> for a particular <Subject>. Each subsequent <Assertion> MUST contain no more than one <AttributeStatement>.

Each <Attribute MUST not be encrypted.

If present, NameFormat of <Attribute> MUST be one of the following:
- urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified
- urn:oasis:names:tc:SAML:2.0:attrname-format:uri
- urn:oasis:names:tc:SAML:2.0:attrname-format:basic

<Attribute> Name MUST be a URI.

Where the definition of an attribute includes one or more descriptors for the attribute, FriendlyName, if present, MUST be one of the defined descriptors.

If <AuthnStatement> accompanies <AttributeStatement>, the following attributes MUST be present and follow the specified format:
- us:gov:e-authentication:basic:assuranceLevel MUST be one of 1, 2, 3, 4, or test and use datatype of xs:string.
- urn:oid:2.5.4.3 MUST have first name followed by optional middle name or initial followed by last name delimited by spaces and MUST be <=256 characters in length and use datatype of xs:string.
- us:gov:e-authentication:basic:specVer MUST be “2.0” for this interface specification and use datatype of xs:string.

These attributes are optional but may provide a richer attribute set for end user.
- urn:oid:2.5.4.4 MUST be <=128 characters in length and of datatype xs:string.
- urn:oid:2.5.4.42 MUST be <=128 characters in length and of datatype xs:string.
- urn:oid:1.3.6.1.4.1.1466.101.120.34 MUST be <=128 characters in length and of datatype xs:string.
- urn:oid:2.5.4.44 MUST be <=20 characters in length and of datatype xs:string.
- us:gov:e-authentication:basic:birthMonth MUST be 2 digits and MUST contain a value in the range of 01 – 12 and of datatype xs:integer.
- us:gov:e-authentication:basic:birthYear MUST be 4 digits (yyyy) and of datatype xs:integer.
- us:gov:e-authentication:basic:address1 MUST be <= 50 characters in length and of datatype xs:string.
- us:gov:e-authentication:basic:address2 MUST be <= 50 characters in length and of datatype xs:string.
- urn:oid:2.5.4.7 MUST be <= 28 characters in length and of datatype xs:string.
- urn:oid:2.5.4.8 MUST be 2 character length state code and of datatype xs:string.
4. SP sends a signed LogoutRequest message to IdP over TLS 1.0 using HTTP Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message over TSL 1.0 using the HTTP Redirect binding.

   <Logout Request>
   
   REQUIRED: Version attribute MUST be set to “2.0”.

   <Logout Response>
   
   REQUIRED: Version attribute MUST be set to “2.0”.

5. SP and IdP configure Common Domain Cookie for IdP Discovery. Domain name will be provided at time of test.

   REQUIRED: SP and IdP have proper cookie for IdP Discovery.

6. User logs in at IdP does Single Sign-On. IdP sends an unsolicited AuthnResponse to SP through HTTP POST binding over TLS to the SP’s <Assertion> Consumer Service. The following list contains either requirements for the request which must occur or optional conditions which may be done.

   <Response>
   
   REQUIRED: Version attribute MUST be set to “2.0”.

   REQUIRED: Each <Response> MUST contain no more than one <EncryptedAssertion>.

   CONDITIONAL: Consent attribute MAY be used, but if used MUST use one of the following:
   
   - urn:oasis:names:tc:SAML:2.0:consent:obtained
   - urn:oasis:names:tc:SAML:2.0:consent:prior
   - urn:oasis:names:tc:SAML:2.0:consent:current-explicit
   - urn:oasis:names:tc:SAML:2.0:consent:unspecified

   <Assertion> element within the <Response>
   
   REQUIRED: An <Assertion> MUST be returned within <EncryptedAssertion> by the IdP and MUST be signed and encrypted.

   REQUIRED: Version MUST be 2.0.

   REQUIRED: <Issuer> MUST be present, MUST be the identifier of the IdP and MUST be a URL reference within the domain of the IdP.

   REQUIRED: There MUST be exactly one <Subject> per <Assertion>.

   <Subject> element within the <Assertion>
   
   REQUIRED: An <Assertion> MUST contain exactly one <Subject> indicating the end user to which <Assertion> pertains.

   REQUIRED: <NameID> MUST contain a Format attribute set either:
<AuthnStatement> element within the <Assertion>

REQUIRED: <AuthnStatement> MUST include the SessionIndex of the end user.

OPTIONAL: <AuthnStatement> MAY contain SessionNotOnOrAfter but SP is NOT REQUIRED to honor SessionNotOnOrAfter.

(AttributeStatement) element within the <Assertion>

REQUIRED: The first transmission of an <Assertion> MUST contain exactly one <AttributeStatement> for a particular <Subject>. Each subsequent <Assertion> MUST contain no more than one <AttributeStatement>.

REQUIRED: Each <Attribute> MUST not be encrypted.

CONDITIONAL: If present, NameFormat of <Attribute> MUST be one of the following:

- urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified
- urn:oasis:names:tc:SAML:2.0:attrname-format:uri
- urn:oasis:names:tc:SAML:2.0:attrname-format:basic

REQUIRED: <Attribute> Name MUST be a URI.

CONDITIONAL: Where the definition of an attribute includes one or more descriptors for the attribute, FriendlyName, if present, MUST be one of the defined descriptors.

CONDITIONAL: If <AuthnStatement> accompanies <AttributeStatement>, the following attributes MUST be present and follow the specified format

- us:gov:e-authentication:basic:assuranceLevel MUST be one of 1, 2, 3, 4, or test and use datatype of xs:string.
- urn:oid:2.5.4.3 MUST have first name followed by optional middle name or initial followed by last name delimited by spaces and MUST be <=256 characters in length and use datatype of xs:string.
- us:gov:e-authentication:basic:specVer MUST be “2.0” for this interface specification and use datatype of xs:string.

CONDITIONAL: These attributes are optional but may provide a richer attribute set for end user.

- urn:oid:2.5.4.4 MUST be <=128 characters in length and of datatype xs:string.
- urn:oid:2.5.4.42 MUST be <=128 characters in length and of datatype xs:string.
- urn:oid:1.3.6.1.4.1.1466.101.120.34 MUST be <=128 characters in length and of datatype xs:string.
- urn:oid:2.5.4.44 MUST be <=20 characters in length and of datatype xs:string.
- us:gov:e-authentication:basic:birthMonth MUST be 2 digits and MUST contain a value in the range of 01 - 12 and of datatype xs:integer.
1188 • us:gov:e-authentication:basic:birthDay MUST be 2 digits and MUST contain a value in
1189 the range of 01 – 31 and of datatype xs:integer.
1190 • us:gov:e-authentication:basic:birthYear MUST be 4 digits (yyyy) and of datatype
1191 xs:integer.
1192 • us:gov:e-authentication:basic:address1 MUST be <= 50 characters in length and of
1193 datatype xs:string.
1194 • us:gov:e-authentication:basic:address2 MUST be <= 50 characters in length and of
1195 datatype xs:string.
1196 • urn:oid:2.5.4.7 MUST be <= 28 characters in length and of datatype xs:string.
1197 • urn:oid:2.5.4.8 MUST be 2 character length state code and of datatype xs:string.
1198 • urn:oid:2.5.4.17 MUST be either 5 digit format or 5digit-4digit (including the dash)
1199 format and of datatype xs:string.
1200 • us:gov:e-authentication:basic:Sid MUST be <= 128 characters in length and of datatype
1201 xs:string.
1202
1203 7. IdP logs out User session. IdP sends a signed LogoutRequest message to SP over TLS 1.0 using
1204 HTTP Redirect binding. SP logs out User session. SP returns a signed LogoutResponse message
1205 over TSL 1.0 using the HTTP Redirect binding.
1206 <Logout Request>
1207     REQUIRED: Version attribute MUST be set to “2.0”.
1208 <Logout Response>
1209     REQUIRED: Version attribute MUST be set to “2.0”.
1210
1211
1212
1213
## References

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
</table>
[GSATechAppr] Dave Silver et al, “Technical Approach for the Authentication Service Component” vs. 2.0.0 GSA (May 2007),

[GSAAAdoptSchm] Dave Silver et al, “E-Authentication Federation Adopted Schemes” vs. 1.0.0 GSA (May 2007),

[GSAIInterface] Dave Silver et al, “E-Authentication Federation Architecture 2.0 Interface Specifications” vs. 1.0.0 GSA (May 2007),
About Drummond Group

Drummond Group Inc. (DGI) is an independent, privately held company that works with software vendors, vertical industries and the standards community to drive adoption for standards by conducting interoperability and conformance testing, publishing related strategic research and developing vertical industry strategies. Founded in 1999, DGI represents best-of-breed in the industry on linking horizontal infrastructure technologies, standards and interoperability issues with the needs of vertical industries such as retail, grocery, health care, transportation, government and automotive. For more information, please visit www.drummondgroup.com or email: info@drummondgroup.com.