Liberty ID-FF 1.2 Static Conformance Requirements
Version: 1.0

Editors:
Eric Tiffany, IEEE-ISTO

Contributors:
John Kemp, IEEE-ISTO

Abstract:
Defines the static conformance requirements for the Liberty Alliance ID-FF version 1.2 specifications.

Filename: liberty-idff-1.2-scr-v1.0.pdf
Notice

This document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to use the document solely for the purpose of implementing the Specification. No rights are granted to prepare derivative works of this Specification. Entities seeking permission to reproduce portions of this document for other uses must contact the Liberty Alliance to determine whether an appropriate license for such use is available.

Implementation of certain elements of this document may require licenses under third party intellectual property rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are not, and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights. This Specification is provided "AS IS", and no participant in the Liberty Alliance makes any warranty of any kind, express or implied, including any implied warranties of merchantability, non-infringement of third party intellectual property rights, and fitness for a particular purpose. Implementors of this Specification are advised to review the Liberty Alliance Project’s website (http://www.projectliberty.org/) for information concerning any Necessary Claims Disclosure Notices that have been received by the Liberty Alliance Management Board.

Copyright © 2004 ActivCard; America Online, Inc.; American Express Travel Related Services; Axalto; Bank of America Corporation; Bell Canada; Cingular Wireless; Cisco Systems, Inc.; Communicator, Inc.; Deloitte & Touche LLP; Earthlink, Inc.; Electronic Data Systems, Inc.; Entrust, Inc.; Epok, Inc.; Ericsson; Fidelity Investments; France Telecom; Gemplus; General Motors; Hewlett-Packard Company; i2 Technologies, Inc.; Internet2; Intuit Inc.; MasterCard International; NEC Corporation; Netegrity, Inc.; NeuStar, Inc.; Nextel Communications; Nippon Telegraph and Telephone Corporation; Nokia Corporation; Novell, Inc.; NTT DoCoMo, Inc.; OneName Corporation; Openwave Systems Inc.; Phaos Technology; Ping Identity Corporation; PricewaterhouseCoopers LLP; RegistryPro, Inc.; RSA Security Inc; Sabre Holdings Corporation; SAP AG; SchlumbergerSema; Sigaba; SK Telecom; Sony Corporation; Sun Microsystems, Inc.; Symlabs, Inc.; Trustgenix; United Airlines; VeriSign, Inc.; Visa International; Vodafone Group Plc; Wave Systems. All rights reserved.
<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>Contents</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>1. Overview</td>
<td>4</td>
</tr>
<tr>
<td>33</td>
<td>2. Conformance Profiles</td>
<td>5</td>
</tr>
<tr>
<td>34</td>
<td>References</td>
<td>31</td>
</tr>
</tbody>
</table>
1. Overview

Static conformance requirements (SCR) describe features that are mandatory and optional for implementations conforming to the Liberty Alliance Identity Federation Framework Specifications (ID-FF version 1.2). This document defines these requirements. This is a normative document with several non-normative explanatory sections.

1.1. Definitions and Motivation

The Liberty specifications define a large number of features and variations. Applications often do not require all the features within a specification. It is also possible that implementations may not be able to implement all the features. In these cases, it may be desirable to partition the specifications into subsets of functionality. A profile is a subset of the overall specifications that includes all of the functionality necessary to satisfy the requirements of a particular community of users. This document identifies several Liberty conformance profiles based on subsets of these specifications according to the following guidelines:

- The number of conformance profiles should be kept small.
- The profiles should correspond to the major roles within the specifications: Identity Provider (IdP), Service Provider (SP), and Liberty Enabled Client/Proxy (LECP).
- The SCR should distinguish between software implementations and deployments. Allow deployments to optionally configure features that are mandatory in the conformance profiles.
- The SCR should place more stringent requirements on IDPs as compared with SPs to promote interoperability.
- Implementations conforming to one (or more) profiles should be able to interoperate with another conforming implementation of a complementary profile.

The resulting profiles are described below by means of tables indicating mandatory and optional features, with references to the appropriate sections of the specification documents.

1.2. Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119]: "they MUST only be used where it is actually required for interoperability or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)."

The normative sections of this document are identified with special formatting as indicated here:

EXAMPLE This is an example of the format of the normative requirements. Each requirement is identified by a requirement identifier ("EXAMPLE" in this example).
2. Conformance Profiles

2.1. Profile Matrices

The following tables summarize the features that comprise the conformance profiles. The detailed specifications for each profile follow in subsequent sections. The first table describes the four profiles derived from the original IDFF 1.1 SCR document. The second table describes the profiles containing the IDFF 1.2 extensions.

### Table 1. Profile Matrix

<table>
<thead>
<tr>
<th>Feature</th>
<th>IDP</th>
<th>SP Basic</th>
<th>SP</th>
<th>LECP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Sign-On using Artifact Profile</td>
<td>MUST</td>
<td>MUST</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Single Sign-On using Browser POST Profile</td>
<td>MUST</td>
<td>MUST</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Single Sign-On using LECP Profile</td>
<td>MUST</td>
<td>MUST</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Register Name Identifier - (IdP Initiated) - HTTP-Redirect</td>
<td>OPTIONAL</td>
<td>MUST</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Register Name Identifier - (IdP Initiated) - SOAP/HTTP</td>
<td>OPTIONAL</td>
<td>OPTIONAL</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Register Name Identifier - (SP Initiated) - HTTP-Redirect</td>
<td>MUST</td>
<td>MUST</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Register Name Identifier - (SP Initiated) - SOAP/HTTP</td>
<td>MUST</td>
<td>OPTIONAL</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Federation Termination Notification (IdP Initiated) – HTTP-Redirect</td>
<td>MUST</td>
<td>MUST</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Federation Termination Notification (IdP Initiated) – SOAP/HTTP</td>
<td>MUST</td>
<td>OPTIONAL</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Federation Termination Notification (SP Initiated) – HTTP-Redirect</td>
<td>MUST</td>
<td>MUST</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Federation Termination Notification (SP Initiated) – SOAP/HTTP</td>
<td>MUST</td>
<td>OPTIONAL</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Single Logout (IdP Initiated) – HTTP-Redirect</td>
<td>MUST</td>
<td>MUST</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Single Logout (IdP Initiated) – HTTP-GET</td>
<td>MUST</td>
<td>MUST</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Single Logout (SP Initiated) – HTTP-Redirect</td>
<td>MUST</td>
<td>OPTIONAL</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Single Logout (SP Initiated) – SOAP</td>
<td>MUST</td>
<td>OPTIONAL</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Identity Provider Introduction (cookie)</td>
<td>MUST</td>
<td>OPTIONAL</td>
<td>OPTIONAL</td>
<td></td>
</tr>
<tr>
<td>Backward Compatibility</td>
<td>OPTIONAL</td>
<td>OPTIONAL</td>
<td>OPTIONAL</td>
<td></td>
</tr>
</tbody>
</table>

The following table summarizes additional profiles which extend the profiles defined above. These extended profiles are to be understood as a combination of an IDP or SP profile from the table above with a corresponding extended profile below.

### Table 2. Extended Profile Matrix

<table>
<thead>
<tr>
<th>Feature</th>
<th>IDP Extended</th>
<th>SP Extended</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Time (Anonymous) Name Identifiers</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Affiliations</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Identity Provider Proxy</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Name Identifier Mapping</td>
<td>MUST</td>
<td>MUST</td>
</tr>
</tbody>
</table>

2.2. Identity Provider Conformance Profile

This section defines the conformance requirements for an identity provider. These requirements are derived from the steps defined in the [LibertyBindProf, Section 3], which describe the interactions between the user agent, the service provider, and the identity provider.

2.2.1. Single Sign-On and Federation
This section defines how an identity provider facilitates single sign-on by processing incoming and outgoing requests. The steps indicated refer to the interaction diagram in [LibertyBindProf, 3.2.1] which illustrates the general single sign-on framework.

### 2.2.1.1. Common Interaction and Processing

The single sign-on requirements specified here assume that the user agent has already authenticated with the identity provider, and that a valid session exists for the user agent at the identity provider.

The following actions are required of the identity provider:

1. **IDP-SSO-001** In step 5, the identity provider MUST process the `<lib:AuthnRequest>` message according to the rules specified in [LibertyProtSchema].

2. **IDP-SSO-002** In step 6, the identity provider MUST respond to the user agent with a `<lib:AuthnResponse>`, a SAML artifact, or an error. The form of this response is contingent on the specific interaction method employed by the identity provider.

3. **IDP-SSO-014** The identity provider MUST return an appropriately scoped `<NameIdentifier>` in accordance with the rules specified for the `<NameQualifier>` in [LibertyProtSchema section 3.2.2.3].

4. **IDP-SSO-015** The identity provider MUST return a `<NameIdentifier>` corresponding to the `<NameIDPolicy>` specified in the authentication request [LibertyProtSchema section 3.2.2.6].

5. **IDP-SSO-016** If the identity provider wishes to proxy authentication to another authentication provider, they MUST follow the rules specified in [LibertyProtSchema section 3.2.2.6] to generate the proxied authentication request.

### 2.2.1.2. Single Sign-on Using Browser Artifact

This section describes the identity provider actions necessary to perform single sign-on using browser artifact. The requirements in this section are derived primarily from Liberty Bindings and Profiles, Section 3.2.2, with additional references to Step 8 in Section 3.1.

1. **IDP-SSO-003** Single Sign-on using Browser Artifact is a mandatory supported feature of the identity provider conformance profile. The requirements in this section MUST be implemented according the the relevant sections of [LibertyBindProf].

2. The identity provider must complete two processing steps to implement this feature: processing an authentication request, and processing a SAML assertion. The authentication interaction proceeds as follows:

3. **IDP-SSO-004** The identity provider MUST process the `<lib:AuthnRequest>` message as specified in [LibertyBindProf, 3.2.1.5, Step 5].

4. **IDP-SSO-005** In response to the `<lib:AuthnRequest>` the identity provider MUST perform a redirection as specified in [LibertyBindProf, 3.2.2.1.2, Step 6].
The identity provider MUST process the `<samlp:Request>` produced by the service provider in Step 8 of the single sign-on interaction, and produce a `<samlp:Response>` as specified in [LibertyBindProf, 3.2.1.9, Step 9].

The artifact produce by the identity provider MUST be formatted as specified in [Liberty-BindProf, 3.2.2.2, Artifact Format].

### 2.2.1.3. Single Sign-on using Liberty Browser POST

This section describes the identity provider requirements for performing single sign-on using Liberty Browser POST.

The identity provider MUST process the `<lib:AuthnRequest>` message as specified in [LibertyBindProf, 3.2.1.5, Step 5]. (Same as [IDP-SSO-004]).

The identity provider generates an HTML 200 OK response containing an authentication request. This response MUST conform to the specification in [LibertyBindProf, 3.2.3.2, Step 6]

### 2.2.1.4. Single Sign-on using Liberty-Enabled Client and Proxy

This section specifies the identity provider requirements for performing single sign-on using the Liberty-Enabled Client and Proxy (LECP) interaction.

The identity provider MUST process the `<lib:AuthnRequest>` in the body of the SOAP POST message from the LECP as specified in [LibertyBindProf, 3.2.1.5, Step 5]. (See [IDP-SSO-004]).

The identity provider MUST respond to the `<lib:AuthnRequest>` with a HTTP 200 OK response as specified in [LibertyBindProf, 3.2.4.2.4, Step 6], with the correct MIME type (`application/vnd.liberty-response+xml`) and Liberty-Enabled HTTP header (see [LibertyBindProf, 3.2.4.1])
2.2.2. Register Name Identifier

This section specifies the required and optional features used by an identity provider to register or change a name identifier for a Principal. There are four variations of the register name identifier protocol: the name registration interaction can be initiated by either the identity provider or the service provider, and the protocol can be either HTTP-Redirect based, or SOAP/HTTP based [LibertyBindProf, 3.3].

Note that while the name registration features are optional interactions in a deployment, some of these features are required for a implementation to be conformant. The following sections describe the mandatory and optional conformance requirements for an identity provider implementing the register name identifier feature.

2.2.2.1. Register Name Identifier Initiated at Identity Provider

2.2.2.1.1. HTTP-Redirect Based

This section specifies the requirements for the HTTP-Redirect-based register name identifier initiated at the identity provider. These requirements are based on the identity provider actions in the interaction described in [LibertyBindProf, 3.3.1.1]. Note that the timing and mechanism of the initiation of this interaction are not normatively specified, although the preceding specification reference offers some examples.

IDP-RNI-001 The identity provider MUST redirect the user agent to the register name identifier service at the service provider as specified in [LibertyBindProf, 3.3.1.1.2, Step 2].

2.2.2.1.2. SOAP/HTTP Based

IDP-RNI-002 The identity provider MUST only initiate SOAP/HTTP-based register name identifier when the service provider metadata specifies the appropriate URI identifier as specified in [LibertyBindProf, 3.3.1.2].

IDP-RNI-003 The SOAP/HTTP-based Register Name Identifier transactions must use the SOAP binding for Liberty as defined in [LibertyBindProf, 2.1].

IDP-RNI-004 The identity provider MUST initiate the Register Name Identifier transaction by sending a <lib:RegisterNameIdentifierRequest> message to the service provider’s SOAP endpoint as specified in [LibertyBindProf, 3.3.1.2.1, Step 1].

IDP-RNI-005 The identity provider MUST process the <lib:RegisterNameIdentifierResponse> from the service provider as specified in [LibertyProtSchema, 3.3.3].

2.2.2.2. Register Name Identifier Initiated at Service Provider

The following sections describe the interactions for an identity provider implementing the Register Name Identifier initiated at the service provider.

IDP-RNI-006 Register Name Identifier initiated at service provider is a REQUIRED feature of the Identity Provider Conformance Profile. Both the HTTP-Redirect and the SOAP/HTTP interaction MUST be implemented.
Note that this section refers to [LibertyBindProf] Register Name Identifier interactions initiated at the identity provider. The steps associated with the service provider in those interactions are used here as if they were associated with the identity provider. All references to service provider and identity provider have been interchanged as indicated in [LibertyBindProf, 3.3.2.1] and [LibertyBindProf, 3.3.2.2].

### 2.2.2.2.1. HTTP-Redirect Based

**IDP-RNI-007** The identity provider MUST process the `<lib:RegisterNameIdentifierRequest>` from the service provider as specified in [LibertyProtSchema, 3.3.3]. See [LibertyBindProf, 3.3.1.1.4, Step 4].

**IDP-RNI-008** The identity provider MUST respond to the service provider with a redirection URL as specified in the `RegisterNameIdentifierServiceReturnURL` metadata element. The redirection MUST adhere to the rules specified in [LibertyBindProf, 3.3.1.1.5, Step 5].

### 2.2.2.2.2. SOAP/HTTP Based

[IDP-RNI-003] is a requirement for this interaction.

**IDP-RNI-009** The service provider will send a `<lib:RegisterNameIdentifierRequest>` protocol message to the identity provider. The identity provider MUST record the new `<lib:SPProvidedNameIdentifier>`.

**IDP-RNI-010** After a successful registration of the `<lib:SPProvidedNameIdentifier>`, the identity provider MUST respond with a `<lib:RegisterNameIdentifierResponse>` according to the processing rules in [LibertyProtSchema, 3.3.3].

### 2.2.3. Identity Federation Termination Notification

Liberty identity federation termination notification specifies how service providers and identity providers are notified of federation termination. There are four variations of the federation termination notification interaction: the federation termination notification interaction can be initiated by either the identity provider or the service provider, and the protocol can be based on either HTTP-Redirect or SOAP/HTTP.

**IDP-FTN-001** This section specifies the conformance requirements for an identity provider to support identity federation termination notification. All four interactions specified in [LibertyBindProf, 3.4] MUST be implemented.

### 2.2.3.1. Federation Termination Notification Initiated at the Identity Provider

#### 2.2.3.1.1. HTTP-Redirect

**IDP-FTN-002** This interaction MUST NOT be used unless the service provider metadata element `FederationTerminationNotificationProtocolProfile` specifies the URI `http://projectliberty.org/profiles/fedterm-idp-http`.

**IDP-FTN-003** This interaction REQUIRES certain preconditions specified in [LibertyBindProf, 3.4.1.1] are met.
In response to a request to the identity provider’s federation termination service URL, the identity provider MUST redirect the user agent to the federation termination service at the service provider. This redirection MUST adhere to the rules specified in [LibertyBindProf, 3.4.1.2, Step 2].

This interaction MUST NOT be used unless the service provider metadata element FederationTerminationNotificationProtocolProfile specifies the URI http://projectliberty.org/profiles/fedterm-idp-soap.

This interaction REQUIRES certain preconditions specified in [LibertyBindProf, 3.4.1.2] are met.

In response to a request from the user agent to the identity provider’s federation termination service URL, the identity provider MUST send an asynchronous SOAP over HTTP notification message to the service provider’s SOAP endpoint. The SOAP message MUST adhere to the rules specified in [LibertyBindProf, 3.4.1.2.2, Step 2].

The service provider will respond to termination notification with a HTTP 204 No Content response.

The identity provider MUST process the HTTP 204 No Content response from the service provider and send an HTTP response confirming the requested action of federation termination with the specified service provider.

The following sections describe the interactions for an identity provider implementing federation termination notification initiated at the service provider.

Note that this section refers to [LibertyBindProf] federation termination notifications interactions initiated at the identity provider. The steps associated with the service provider in those interactions are used here as if they were associated with the identity provider. All references to to service provider and identity provider have been interchanged as indicated in [LibertyBindProf, 3.4.2.1] and [LibertyBindProf, 3.4.2.2].

The identity provider MUST process the <lib:FederationTerminationNotification> received from the user agent according to the rules defined in [LibertyProtSchema, 3.4.2] and in [LibertyBindProf, 3.4.1.1.4, Step 4].

The identity provider’s federation termination service MUST respond by redirecting the user agent as specified in [LibertyBindProf, 3.4.1.1.5, Step 5].

The identity provider MUST process the <lib:FederationTerminationNotification> in the SOAP message received from the service provider according to the rules defined in [LibertyProtSchema, 3.4.2] and in [LibertyBindProf, 3.4.1.2.3, Step 3].
The identity provider MUST respond to the `<lib:FederationTerminationNotification>` with a HTTP 204 OK response [LibertyBindProf, 3.4.1.2.4, Step 4].

2.2.4. Single Logout

Liberty single logout specifies how service providers and identity providers synchronize logout across all sessions authenticated by a particular identity provider. There are five variations of the single logout interaction: the single logout can be initiated by either the identity provider or the service provider, and the protocol can be based on either HTTP-Redirect, HTTP-GET (only when initiated at the identity provider), or SOAP/HTTP.

This section specifies the conformance requirements for an identity provider to support Single Logout. All five interactions specified in [LibertyBindProf, 3.5] MUST be implemented.

2.2.4.1. Single Logout Initiated at the Identity Provider

The following sections specify the requirements for single logout when initiated by a user agent at the identity provider.

2.2.4.1.1. HTTP-Redirect

This interaction MUST NOT be used unless the service provider metadata element `SingleLogoutProtocolProfile` specifies the URI `http://projectliberty.org/profiles/slo-idp-http`.

In response to the user agent request, the identity provider MUST redirect the user agent to the single logout service URL at each service provider for which the identity provider has provided an authentication assertion during the Principal’s current session. Each redirection MUST adhere to the rules specified in [LibertyBindProf, 3.5.1.1.1.2, Step 2].

After receiving the request from the user agent to the `SingleLogoutServiceReturnURL` as specified in the identity provider metadata, the identity provider MUST process the request and send an HTTP response to the user agent confirming that the requested action of a single logout has been completed.

2.2.4.1.2. HTTP-GET

This interaction MUST NOT be used unless the service provider metadata element `SingleLogoutProtocolProfile` specifies the URI `http://projectliberty.org/profiles/slo-idp-http`.

In response to the user agent request, the identity provider MUST respond with a standard HTTP 200 OK response containing image tags referencing the logout service URL for each of the service providers for which the identity provider has provided an authentication assertion during the Principal’s current session. Each image tag MUST adhere to the rules specified in [LibertyBindProf, 3.5.1.1.2.2, Step 2].
After receiving the request from the user agent to the `SingleLogoutServiceReturnURL` as specified in the identity provider metadata, the identity provider MUST process the request and send an HTTP response to the user agent confirming that the requested action of a single logout has been completed.

**2.2.4.1.3. SOAP/HTTP**

The requirements for the SOAP/HTTP single logout interaction are described in [LibertyBindProf, 3.5.1.2].

This interaction MUST NOT be used unless the service provider metadata element `SingleLogoutProtocolProfile` specifies the URI `http://projectliberty.org/profiles/slo-idp-soap`.

The identity provider MUST send a SOAP over HTTP request to the SOAP endpoint of each service provider for which the identity provider provided authentication assertions during the Principal’s current session. Each message MUST contain exactly one `<lib:LogoutRequest>` element in the SOAP body and adhere to the rules specified in [LibertyBindProf, 3.5.1.2.2] and [LibertyProtSchema, 3.5.1].

In response to a SOAP 200 OK `<lib:LogoutResponse>` message from the service provider, the identity provider MUST send an HTTP response confirming the requested action of single logout has completed.

**2.2.4.2. Single Logout Initiated at the Service Provider**

**2.2.4.2.1. HTTP-Redirect**

The user agent will access the identity provider’s single logout service URL. The identity provider MUST process the `<lib:LogoutRequest>` according to the rules defined in [LibertyProtSchema, 3.5.1].

The identity provider MUST notify each service provider for which the identity provider has provided authentication assertions of the logout request via the service provider’s preferred profile [LibertyBindProf, 3.5.2.1.4, Step 4].

The identity provider MUST terminate the Principal’s current session, and no more authentication assertions for the Principal are to be given to service providers.

The identity provider MUST respond and redirect the user agent back to the service provider using the return URL location obtained from the `SingleLogoutServiceReturnURL` metadata element as specified in [LibertyBindProf, 3.5.2.1.5, Step 5].

After receiving a `<lib:LogoutRequest>` from the service provider, the identity provider MUST process it according to the rules in [LibertyProtSchema, 3.5.1].
IDP-SLO-015  The identity provider MUST submit to each service provider for which the identity provider has provided authentication assertions during the Principal’s current session a request to logout the Principal as specified in [LibertyBindProf, 3.5.2.2.3, Step 3].

IDP-SLO-016  The identity provider MUST respond to the <lib:LogoutRequest> with a SOAP 200 OK <lib:LogoutResponse> message [LibertyBindProf, 3.5.2.2.4, Step 4].

2.2.5. Identity Provider Introduction

This section describes the conformance requirements for an identity provider implementing the identity provider introduction feature. The identity provider introduction feature is intended to allow service providers to discover which identity providers a Principal is using.

IDP-IPI-001  The identity provider introduction feature is a REQUIRED element of identity provider conformance profile.

Although a deployment may choose not to enable the identity provider introduction feature, an identity provider implementation must provide the feature in order to be conformant.

2.2.5.1. Common Domain Cookie

The identity provider introduction relies on the use of a common domain cookie.

IDP-IPI-002  The common domain cookie MUST be constructed as specified in [LibertyBindProf, 3.6.1].

2.2.5.2. Setting the Common Domain Cookie

Creating and updating the common domain cookie is the responsibility of the identity provider.

IDP-IPI-003  After authenticating a Principal, the identity provider MUST attempt to set the common domain cookie, subject to cookie-setting restrictions of the user-agent.

The details of this procedure are implementation-dependent, and are not normatively specified. However, one possible strategy is described in [LibertyBindProf, 3.6.2].

2.2.6. Backward Compatibility with v1.1 Protocol

Backward compatibility is optional for conformant identity providers:

IDP-BCK-001  Backward compatibility with 1.1 service provider implementations is OPTIONAL for an identity provider implementation. However, an implementation claiming backward compatible conformance MUST adhere to the relevant rules as described in [LibertyProtSchema, 3.1.11].

2.3. Extended Identity Provider Conformance Profile

This section describes the requirements for an implementation to be an Extended Identity Provider. This profile extends the Identity Provider Profile defined in the previous section.

IDP-EXT-001  An Extended Identity Provider implementation MUST conform to the Identity Provider profile requirements. In addition, an Extended Identity Provider MUST conform to all the requirements defined in the following subsections.
2.3.1. One-Time (Anonymous) Name Identifier

A service provider can request a one-time (also known as "anonymous") name identifier for a Principal during the sign-on interaction with the identity provider by specifying a value of `onetime` for the NameIDPolicy element in the AuthnRequest.

In this case, the identity provider will construct the AuthnResponse according to the following requirements:

| IDP-ONE-001 | If the `<NameIDPolicy>` element is `onetime`, then the `<saml:NameIdentifier>` element in the `<saml:Subject>` element MUST be a temporary, one-time-use identifier for the Principal, with a Format attribute of `urn:liberty:iff:nameid:one-time`. |
| IDP-ONE-002 | The temporary identifier MUST be constructed according to the specifications in [LibertyProtSchema, 3.1.4]. |

2.3.2. Affiliations

An affiliation is a set of one or more entities, described by providerIDs, who may perform Liberty interactions as a member of the set. This section describes the conformance requirements for an identity provider implementing the affiliation features of the IDFF 1.2 specifications. These requirements are derived from protocol processing rules in [LibertyProtSchema].

| IDP-AFF-001 | The value of the `<AffiliationID>` element of ID-FF 1.2 protocol messages MUST adhere to the uniqueness constraints described in [LibertyProtSchema, 3.1.3]. |

2.3.2.1. Single Sign-on and <AuthnResponse>

The identity provider will issue an `<AuthnResponse>` after processing a `<AuthnRequest>` from a service provider (or, in some cases, an identity provider may issue an `<AuthnResponse>` without first receiving such a request). These are the requirements that obtain when an affiliation is present in the request (see [LibertyProtSchema, 3.2.2.6]).

| IDP-AFF-002 | If the `<AffiliationID>` element is present, then the `<saml:NameIdentifier>` MUST be the most recent name identifier provided by a member of the affiliation, if any, or the name identifier for the Principal supplied by the identity provider for the affiliation. |
| IDP-AFF-003 | `<AffiliationID>`, if present, MUST be the unique identifier of a known affiliation group with which the identity provider has an established relationship, and of which the requesting provider is a member. If present, identity providers MUST establish and resolve federations based on the specified affiliation, not the requesting provider. In addition, identity providers MAY retrieve information regarding the other members of the affiliation group by querying metadata (see [LibertyMetadata]) and present a list of members of the affiliation group to the Principal. |
| IDP-AFF-004 | In all of the name identifier elements in the request and response messages of this protocol, if the Principal’s identity federation is between the identity provider and an affiliation group in which the service provider is a member, then the NameQualifier attribute MUST contain the unique identifier of the affiliation group. Otherwise, it MUST contain the unique identifier of the service provider. This attribute MUST be used by the providers to identify the specific identity federation to be modified. |
2.3.2.2. Name Registration

The provider is required to correctly identify the affiliation in the <RegisterNameIdentifierRequest> and <RegisterNameIdentifierResponse> messages.

IDP-AFF-005 In all of the name identifier elements in the request and response messages of this protocol, if the Principal’s identity federation is between the identity provider and an affiliation group in which the service provider is a member, then the NameQualifier attribute MUST contain the unique identifier of the affiliation group. Otherwise, it MUST contain the unique identifier of the service provider. This attribute MUST be used by the providers to identify the specific identity federation to be modified.

See [LibertyProtSchema, 3.3.3].

2.3.2.3. Federation Termination

The provider is required to correctly identify the affiliation when performing federation termination using the <FederationTerminationNotification> message.

IDP-AFF-006 If the Principal’s identity federation was between the identity provider and an affiliation group in which the service provider is a member, then the NameQualifier attribute MUST contain the unique identifier of the affiliation group. Otherwise, it MUST contain the unique identifier of the service provider. This attribute MUST be used by the providers to identify the specific identity federation being terminated.

See [LibertyProtSchema, 3.4.2].

2.3.2.4. Single Logout

The provider is required to correctly identify the affiliation when responding to a single logout request using the <LogoutResponse> message.

IDP-AFF-007 If the Principal’s identity federation is between the identity provider and an affiliation group in which the service provider is a member, then the NameQualifier attribute MUST contain the unique identifier of the affiliation group. Otherwise, it MUST contain the unique identifier of the service provider. This attribute MUST be used by the providers to identify the specific identity federation of the Principal who is logging out.

See [LibertyProtSchema, 3.5.2.3].

2.3.3. Dynamic Proxy of Identity Provider

A service provider may request that an identity provider obtain authentication for a Principal from a third-party identity provider. This section describes the conformance requirements for an identity provider implementing this dynamic proxy feature.

IDP-PXY-001 A conformant identity provider implementation MUST adhere to the processing rules defined in [LibertyProtSchema, 3.2.2.7].

However, note that the dynamic proxy specifications do not compel an identity provider to perform a proxy interaction. For purposes of conformance, the following directives are made compulsory. In an actual deployment, whether or not to proxy would be a local policy issue.
The identity provider MUST proxy an authentication request if the value in the `<ProxyCount>` element is greater than zero, or if no `<ProxyCount>` appears in the request.

The identity provider MUST proxy for a provider specified in the `<IDPList>`.

### 2.3.4. Name Identifier Mapping

This section describes the conformance requirements for an identity provider performing NameIdentifier mapping. The requirements in this section are derived from [LibertyBindProf, 3.7]. The SOAP-based NameIdentifier interaction is the sole interaction described in the specification.

- **IDP-NIM-001** The SOAP-based NameIdentifier Mapping interaction is a mandatory supported feature of the Extended Identity Provider Conformance Profile. The requirements in this section MUST be implemented according to the relevant section of [LibertyBindProf).

The NameIdentifier mapping interaction specifies how one service provider (the "requester") may obtain a NameIdentifier for a Principal it has federated in the namespace of a different service provider (the "target"). The "requester" accomplishes this by issuing a `<lib:NameIdentifierMappingRequest>` to an identity provider that has federated the Principal with both service providers (see [LibertyBindProf, Figure 17 and 3.7.1.1]).

- **IDP-NIM-002** The identity provider MUST process the `<lib:NameIdentifierMappingRequest>` as indicated in [LibertyBindProf, 3.7.1.2, Step 2] and in [LibertyProtSchema, 3.6.2 and 3.6.3]. Specifically:

  - **IDP-NIM-003** The identity provider MUST validate any signature present on the message. The signature MUST be the signature of the `<ProviderID>` contained in the message. If the signature is invalid, the identity provider MUST ignore the message.

  - **IDP-NIM-004** The identity provider MUST respond to the `<lib:NameIdentifierMappingRequest>` with a SOAP 200 OK `<lib:NameIdentifierMappingResponse>` message. The identity provider is NOT required to honor the request for a mapped NameIdentifier, but it MUST respond to the request with an appropriate status.

  - **IDP-NIM-005** The identity provider MUST construct the `<lib:NameIdentifierMappingResponse>` in accordance with the rules described in [LibertyProtSchema, 3.6.3.1].

  - **IDP-NIM-006** The identity provider SHOULD encrypt or otherwise obfuscate the NameIdentifier returned to the requesting service provider.

### 2.3.4.1. Encrypted Name Identifiers

This section describes the requirements an identity provider must meet when encrypting and encoding a NameIdentifier, such as might be needed to satisfy [IDP-NIM-006].

- **IDP-ENI-001** Encrypted Name Identifiers are a mandatory supported feature of the Extended Identity Provider Conformance Profile. The requirements in this section MUST be implemented according to the relevant section of [LibertyBindProf, 3.8].
IDP-ENI-002 This interaction MUST NOT be used unless the provider metadata element NameIdentifierMappingEncryptionProfile specifies the URI urn:liberty:iff:nameid:encrypted.

2.3.4.1. XML Encrypting and Encoding

The <EncryptableNameIdentifier> element is used to contain the encrypted data.

IDP-ENI-003 The identity provider MUST transform the original <saml:NameIdentifier> element into a <EncryptableNameIdentifier> element, encrypt and encode it as described in [Liberty-BindProf, 3.8.1.1].

The requirements in this section rely on the use of [xmlenc-core] to format and encode the encrypted data. Note that certain block encryption algorithms are mandatory for use with the <xenc:EncryptedData> and therefore must be supported:

IDP-ENI-004 The following algorithms MUST be supported as indicated in [xmlenc-core], sections 5.2.1 and 5.2.2: TRIPLE DES, AES-128, AES-256.

2.3.4.1.2. Security Considerations

The security considerations surrounding the Encrypted NameIdentifier are described in [LibertyBindProf, 3.8.2].

IDP-ENI-005 The identity provider MUST follow the requirements in [LibertyBindProf, 3.8.2] regarding the handling of symmetric encryption keys.

2.4. Service Provider Basic Conformance Profile

This section defines the minimal conformance requirements for a Service Provider. These requirements are derived from the steps defined in the [LibertyBindProf, Section 3], which describe the interactions between the user agent, the service provider, and the identity provider.

2.4.1. Single Sign-On and Federation

2.4.1.1. Common Interaction and Processing

The single sign-on requirements specified here assume that the user agent has already authenticated with the identity provider, and that a valid session exists for the user agent at the identity provider.

2.4.1.2. Single Sign-on using Browser Artifact

This section describes the service provider actions necessary to perform single sign-on using browser artifact.

SP-SSO-001 Single sign-on using browser artifact is a mandatory feature of the service provider basic conformance profile. The requirements in this section MUST be implemented as specified in [LibertyBindProf, Section 3.2.2] and [LibertyBindProf, 3.2.1, Steps 8 and 10].

The user agent initiates the single sign-on by making an HTTP request to the service provider, as indicated in [LibertyBindProf, 3.2.1.1, Step 1]. The service provider then obtains the address of the appropriate identity provider in an implementation-dependent way (not normatively specified).
The service provider MAY use the Liberty identity provider introduction interaction while processing the user agent request. After determining the appropriate identity provider location, the service provider then responds to the user agent with an HTTP redirect.

The HTTP redirection response MUST adhere to the specifications in [LibertyBindProf, 3.2.2.1, Step 3]. After obtaining an SAML assertion from the identity provider, the user agent accesses the assertion consumer service at the service provider.

The service provider MUST send a <samlp:Request> SOAP message to the identity provider’s SOAP endpoint as specified in [LibertyBindProf, 3.2.2.1, Step 8].

The service provider MUST process the <saml:Assertion> returned by the identity provider as specified in [LibertyBindProf, 3.2.2.1, Step 10].

This section describes the service provider requirements for performing single sign-on using Liberty browser POST. Single sign-on using Liberty browser POST is a mandatory supported feature of the service provider Basic conformance profile. The requirements in this section MUST be implemented according to the relevant sections of [LibertyBindProf, 3.2.3].

The user agent initiates the single sign-on by making an HTTP request to the service provider’s intersite transfer service as indicated in [LibertyBindProf, 3.2.1.1, Step 1]. The service provider then obtains the address of the appropriate identity provider in an implementation-dependent way (not normatively specified).

The service provider’s responds to the user agent request by redirecting the user agent to the single sign-on service URL at the identity provider.

The redirection MUST adhere to the rules specified in [LibertyBindProf, 3.2.3, Step 3]. After obtaining an authentication assertion, the user agent will issue an HTTP POST request containing the <lib:AuthnResponse> to the service provider.

The service provider MUST process the <lib:AuthnResponse> in the HTTP POST from the user agent as specified in [LibertyBindProf, 3.2.2.1, Step 10].

This section specifies the service provider requirements for performing single sign-on using the Liberty-Enabled Client and Proxy (LECP) interaction.

Single sign-on using LECP is a mandatory supported feature of the service provider basic conformance profile. The requirements in this section MUST be implemented according to [LibertyBindProf, 3.2.4].

The user agent will submit request to the service provider which contains the requisite Liberty-Enabled indications.
SP-SSO-010  The service provider MUST NOT obtain an identity provider address or perform identity provider introduction.

SP-SSO-011  The service provider MUST issue an HTTP 200 OK response to the user agent. The response MUST adhere to the specifications of [LibertyBindProf, 3.2.4.2, Step 3].

SP-SSO-012  If the service provider does not support Liberty version advertised by the LECP, the service provider MUST return an HTTP 501 Not Implemented response to the LECP with the reason phrase "Unsupported Liberty Version."

SP-SSO-013  The service provider SHOULD place appropriate headers in the response to ensure the response is not cached as specified in [LibertyBindProf, 3.2.4.2, Step 3].

SP-SSO-014  The service provider MUST process the <lib:AuthnResponse> in the HTTP POST from the user agent as specified in [LibertyBindProf, 3.2.2.1, Step 10]

2.4.2. Register Name Identifier

2.4.2.1. Register Name Identifier Initiated at Identity Provider

The following sections describe the interactions for a service provider implementing the register name identifier initiated at the identity provider.

2.4.2.1.1. HTTP-Redirect Based

SP-RNI-001  The HTTP-Redirect based register name identifier (initiated at the identity provider) is a REQUIRED feature of the service provider basic profile.

SP-RNI-002  The service provider MUST process the <lib:RegisterNameIdentifierRequest> from the identity provider as specified in [LibertyProtSchema, 3.3.3]. See [LibertyBindProf, 3.3.1.1.4, Step 4].

SP-RNI-003  The service provider MUST respond to the identity provider with a redirection URL as specified in the RegisterNameIdentifierServiceReturnURL metadata element. The redirection MUST adhere to the rules specified in [LibertyBindProf, 3.3.1.1.5, Step 5].

2.4.2.1.2. SOAP/HTTP Based

SP-RNI-004  The SOAP/HTTP based register name identifier (initiated at the identity provider) is an OPTIONAL feature of the service provider basic profile.

SP-RNI-005  The SOAP/HTTP-based register name identifier transactions MUST use the SOAP binding for Liberty as defined in [LibertyBindProf, 2.1].
Liberty Alliance Project: Liberty ID-FF 1.2 Static Conformance Requirements

Version: 1.0

The identity provider will send a `<lib:RegisterNameIdentifierRequest>` protocol message to the service provider. The service provider MUST record the new `<lib:IDPProvidedNameIdentifier>`.

After a successful registration of the `<lib:IDPProvidedNameIdentifier>`, the service provider MUST respond with a `<lib:RegisterNameIdentifierResponse>` according to the processing rules in [LibertyProtSchema, 3.3.3].

2.4.2.2. Register Name Identifier Initiated at Service Provider

Note that this section refers to [LibertyBindProf] register name identifier interactions initiated at the identity provider. The steps associated with the service provider in those interactions are used here as if they were associated with the identity provider. All references to service provider and identity provider have been interchanged as indicated in [LibertyBindProf, 3.3.2.2].

2.4.2.2.1. HTTP-Redirect Based

The HTTP-Redirect based register name identifier (initiated at the service provider) is a REQUIRED feature of the service provider basic profile.

The service provider can initiate the register name identifier interaction, though the circumstances of this initiation are not normatively specified.

The service provider MUST redirect the user agent to the register name identifier service at the identity provider as specified in [LibertyBindProf, 3.3.1.1.2, Step 2].

2.4.2.2.2. SOAP/HTTP Based

The SOAP/HTTP based register name identifier (initiated at the service provider) is an OPTIONAL feature of the service provider basic profile.

The service provider MUST only initiate SOAP/HTTP-based register name identifier when the identity provider metadata specifies the appropriate URI identifier as specified in [LibertyBindProf, 3.3.1.2].

The SOAP/HTTP-based register name identifier transactions MUST use the SOAP binding for Liberty as defined in [LibertyBindProf, 2.1].

The service provider MUST initiate the register name identifier transaction by sending a `<lib:RegisterNameIdentifierRequest>` message to the identity provider’s SOAP endpoint as specified in [LibertyBindProf, 3.3.1.2, Step 1].

The service provider MUST process the `<lib:RegisterNameIdentifierResponse>` from the identity provider as specified in [LibertyProtSchema, 3.3.3].
2.4.3. Identity Federation Termination Notification

Liberty identity federation termination notification specifies how service providers and identity providers are notified of federation termination. There are four variations of the federation termination notification interaction: the federation termination notification interaction can be initiated by either the identity provider or the service provider, and the protocol can be based on either HTTP-Redirect or SOAP/HTTP.

2.4.3.1. Federation Termination Notification Initiated at the Identity Provider

2.4.3.1.1. HTTP-Redirect

SP-FTN-001 The HTTP-Redirect based federation termination notification (initiated at the identity provider) is a REQUIRED feature of the service provider basic profile.

SP-FTN-002 The service provider MUST process the `<lib:FederationTerminationNotification>` received from the user agent according to the rules defined in [LibertyProtSchema, 3.4.2] and in [LibertyBindProf, 3.4.1.1.4, Step 4].

SP-FTN-003 The service provider’s federation termination service MUST respond by redirecting the user agent as specified in [LibertyBindProf, 3.4.1.1.5, Step 5].

2.4.3.1.2. SOAP/HTTP

SP-FTN-004 The SOAP/HTTP based federation termination notification (initiated at the identity provider) is an OPTIONAL feature of the service provider basic profile.

SP-FTN-005 The service provider MUST process the `<lib:FederationTerminationNotification>` in the SOAP message received from the identity provider according to the rules defined in [LibertyProtSchema, 3.4.2] and in [LibertyBindProf, 3.4.1.2.3, Step 3].

SP-FTN-006 The service provider MUST respond to the `<lib:FederationTerminationNotification>` with a HTTP 204 OK response [LibertyBindProf, 3.4.1.2.4, Step 4].

2.4.3.2. Federation Termination Initiated at the Service Provider

2.4.3.2.1. HTTP-Redirect

SP-FTN-007 The HTTP-Redirect based federation termination notification (initiated at the service provider) is a REQUIRED feature of the service provider basic profile.

SP-FTN-008 This interaction MUST NOT be used unless the identity provider metadata element `FederationTerminationNotificationProtocolProfile` specifies the URI `http://projectliberty.org/profiles/fedterm-sp-http`.

SP-FTN-009 This interaction REQUIRES certain preconditions specified in [LibertyBindProf, 3.4.1.1] are met.
In response to a request to the service provider’s federation termination service URL, the service provider MUST redirect the user agent to the federation termination service at the identity provider. This redirection MUST adhere to the rules specified in [LibertyBindProf, 3.4.1.1.2, Step 2].

2.4.3.2.2. SOAP/HTTP

The SOAP/HTTP based federation termination notification (initiated at the service provider) is an OPTIONAL feature of the service provider basic profile.

This interaction MUST NOT be used unless the identity provider metadata element `FederationTerminationNotificationProtocolProfile` specifies the URI `http://projectliberty.org/profiles/fedterm-sp-soap`.

This interaction REQUIRES certain preconditions specified in [LibertyBindProf, 3.4.1.2] are met.

In response to a request from the user agent to the service provider’s federation termination service URL, the service provider MUST send an asynchronous SOAP over HTTP notification message to the identity provider’s SOAP endpoint. The SOAP message MUST adhere to the rules specified in [LibertyBindProf, 3.4.1.2.2, Step 2].

The identity provider will respond to termination notification with a HTTP 204 No Content response.

The service provider MUST process the HTTP 204 No Content response from the identity provider and send an HTTP response confirming the requested action of federation termination with the specified identity provider.

2.4.4. Single Logout

Liberty single logout specifies how service providers and identity providers synchronize logout across all sessions authenticated by a particular identity provider. There are five variations of the single logout interaction: the single logout can be initiated by either the identity provider or the service provider, and the protocol can be based on either HTTP-Redirect, HTTP-GET (only when initiated at the identity provider), or SOAP/HTTP.

Note that Single Logout, in the general case, is an iterative process from the perspective of an identity provider since the identity provider must contact each service provider to which it has issued authentication assertions. However, for a service provider, the single logout interaction is a single event.

2.4.4.1. Single Logout Initiated at the Identity Provider

2.4.4.1.1. HTTP-Redirect

The HTTP-Redirect based single logout interaction (initiated at the identity provider) is a REQUIRED feature of the service provider basic profile.
SP-SLO-002 The user agent will access the service provider’s single logout service URL via a redirect from the identity provider. The service provider MUST process the `<lib:LogoutRequest>` according to the rules defined in [LibertyProtSchema, 3.5.1].

SP-SLO-003 The service provider MUST invalidate the session(s) of the Principal referred to in the name identifier received from the identity provider in the `<lib:LogoutRequest>`.

SP-SLO-004 The service provider MUST respond and redirect the user agent back to the identity provider using the return URL location obtained from the `SingleLogoutServiceReturnURL` metadata element as specified in [LibertyBindProf, 3.5.1.1.1.5, Step 5].

2.4.4.1.2. HTTP-GET

SP-SLO-005 The HTTP-GET based single logout interaction (initiated at the identity provider) is a REQUIRED feature of the service provider basic profile. The user agent will access the single logout service URL of the service provider as a result of an image tag load generated by the identity provider.

SP-SLO-006 The service provider MUST process the `<lib:LogoutRequest>` according to the rules defined in [LibertyProtSchema, 3.5.1].

SP-SLO-007 The service provider MUST invalidate the session(s) of the Principal referred to in the name identifier received from the identity provider in the `<lib:LogoutRequest>`.

SP-SLO-008 The service provider MUST respond and redirect the user agent image load back to the identity provider’s logout completion URL obtained from the `SingleLogoutServiceReturnURL` metadata element. The HTTP response MUST be formed as specified in [LibertyBindProf, 3.5.1.1.2.5, Step 5].

2.4.4.1.3. SOAP/HTTP

SP-SLO-009 The SOAP/HTTP based single logout interaction (initiated at the identity provider) is an OPTIONAL feature of the service provider basic profile.

SP-SLO-010 After receiving a `<lib:LogoutRequest>` from the identity provider, the service provider MUST process it according to the rules in [LibertyProtSchema, 3.5.1].

SP-SLO-011 The service provider MUST invalidate the session(s) of the Principal referred to in the name identifier received from the identity provider in the `<lib:LogoutRequest>`.

SP-SLO-012 The service provider MUST respond to the `<lib:LogoutRequest>` with a SOAP 200 OK `<lib:LogoutResponse>` message [LibertyBindProf, 3.5.1.2, Step 4].
2.4.4.2. Single Logout Initiated at the Service Provider

2.4.4.2.1. HTTP-Redirect

SP-SLO-013  The HTTP-Redirect based single logout interaction (initiated at the service provider) is a REQUIRED feature of the service provider basic profile.

The user agent will access the single logout service URL at the service provider.

SP-SLO-014  The service provider’s single logout service responds and redirects the user agent to the single logout service at the identity provider. The HTTP redirect MUST adhere to the rules specified in [LibertyBindProf, 3.5.2.1.2, Step 2].

After the identity provider has processed the single logout request and contacted the appropriate service providers, the user agent will be redirected back to the service provider contacted originally.

SP-SLO-015  The service provider SHOULD send an HTTP 200 OK response to the user agent with confirmation of the logout.

2.4.4.2.2. SOAP/HTTP

SP-SLO-016  The SOAP/HTTP based single logout interaction (initiated at the service provider) is an OPTIONAL feature of the service provider basic profile.

The user agent will initiate single logout by accessing the single logout service URL at the service provider via an HTTP request.

SP-SLO-017  In response to the single logout request, the service provider sends a SOAP over HTTP request to the identity provider’s SOAP endpoint. The SOAP request MUST be constructed and processed as specified in [LibertyBindProf, 3.5.2.2, Step 2].

The identity provider will contact all service providers to which it has issued assertions for the Principal to request a logout action. The identity provider may determine that one or more of the service providers do not support the SOAP single logout interaction. The identity provider will return a <lib:LogoutResponse> containing a status code of <lib:UnsupportedProfile>

SP-SLO-018  If the identity provider responds to the single logout request with <lib:UnsupportedProfile>, the service provider MUST re-submit its <lib:LogoutRequest> via the HTTP interaction specified above.

SP-SLO-019  If the identity provider responds to the logout request with a SOAP 200 OK <lib:LogoutResponse>, indicating successful single logout, the service provider SHOULD send a HTTP response to the user agent confirming the single logout.

2.4.5. Identity Provider Introduction

This section describes the conformance requirements for a service provider implementing the identity provider introduction feature. The identity provider introduction feature is intended to allow service providers to discover which identity providers a Principal is using.

SP-IPI-001  The identity provider introduction feature is an OPTIONAL element of the service provider basic conformance profile.
2.4.5.1. Common Domain Cookie

The identity provider introduction relies on the use of a common domain cookie.

SP-IPI-002 The common domain cookie MUST be constructed as specified in [LibertyBindProf, 3.6.1].

2.4.5.2. Obtaining the Common Domain Cookie

The service provider uses the common domain cookie to determine which identity providers a Principal uses. The common domain cookie is presented to the service provider after being read by an HTTP server in the common domain; the details of this interaction are outside the scope of this document.

SP-IPI-003 If the HTTP server in the common domain is operated by the service provider, the service provider MAY redirect the user agent to an identity provider for single sign-on.

The details of this procedure are implementation-dependent, and are not normatively specified. However, one possible strategy is described in [LibertyBindProf, 3.6.3].

2.4.6. Backward Compatibility

Backward compatibility is optional for conformant service providers:

SP-BCK-001 Backward compatibility with 1.1 identity provider implementations is OPTIONAL for an service provider implementation. However, an implementation claiming backward compatible conformance MUST adhere to the relevant rules as described in [LibertyProtSchema, 3.1.11].

However, note that only required conformance features need interoperate:

SP-BCK-002 Conformant service provider implementations are NOT REQUIRED to be backward compatible with conformance features described as optional.

In other words, Service Provider Basic implementations need only be backward compatible for protocol features that are required as part of the Service Provider Basic conformance profile.

2.5. Service Provider Conformance Profile

This section defines the conformance requirements for the Service Provider Conformance Profile.

SP-CONF The service provider profile is defined to be the service provider basic profile with all optional interactions changed to REQUIRED except for the identity provider introduction interaction.

2.6. Extended Service Provider Conformance Profile

This section describes the requirements for an implementation to be an Extended Service Provider. This profile extends the Service Provider Profile defined in the previous section.

SP-EXT-001 An Extended Service Provider implementation MUST conform to the Service Provider profile requirements. In addition, an Extended Service Provider MUST conform to all the requirements defined in the following subsections.
2.6.1. One-Time (Anonymous) Name Identifier

A service provider can request a one-time (also known as "anonymous") name identifier for a Principal during the sign-on interaction with the identity provider.

SP-ONE-001 To request a one-time name identifier a service provider MUST specify a value of onetime for the NameIDPolicy element in the AuthnRequest.

2.6.2. Affiliations

An affiliation is a set of one or more entities, described by providerIDs, who may perform Liberty interactions as a member of the set. This section describes the conformance requirements for an service provider implementing the affiliation features of the ID-FF 1.2 specifications. These requirements are derived from protocol processing rules in [LibertyProtSchema].

SP-AFF-001 The value of the <AffiliationID> element of ID-FF 1.2 protocol messages MUST adhere to the uniqueness constraints described in [LibertyProtSchema, 3.1.3].

2.6.2.1. Single Sign-on and <AuthnRequest>

The service provider will issue an <AuthnRequest> to initiate single sign-on. These are the requirements that obtain when an affiliation is present in the request (see [LibertyProtSchema, 3.2.2.6]).

SP-AFF-002 If the <AffiliationID> element is present, then the <saml:NameIdentifier> MUST be the most recent name identifier provided by a member of the affiliation, if any, or the name identifier for the Principal supplied by the identity provider for the affiliation.

SP-AFF-003 <AffiliationID>, if present, MUST be the unique identifier of a known affiliation group with which the identity provider has an established relationship, and of which the requesting provider is a member.

2.6.2.2. Name Registration

The provider is required to correctly identify the affiliation in the <RegisterNameIdentifierRequest> and <RegisterNameIdentifierResponse> messages.

SP-AFF-004 In all of the name identifier elements in the request and response messages of this protocol, if the Principal’s identity federation is between the identity provider and an affiliation group in which the service provider is a member, then the NameQualifier attribute MUST contain the unique identifier of the affiliation group. Otherwise, it MUST contain the unique identifier of the service provider. This attribute MUST be used by the providers to identify the specific identity federation to be modified.
2.6.2.3. Federation Termination

The provider is required to correctly identify the affiliation when performing federation termination using the `<FederationTerminationNotification>` message.

- **SP-AFF-005**: If the Principal’s identity federation was between the identity provider and an affiliation group in which the service provider is a member, then the `NameQualifier` attribute MUST contain the unique identifier of the affiliation group. Otherwise, it MUST contain the unique identifier of the service provider. This attribute MUST be used by the providers to identify the specific identity federation being terminated.

2.6.2.4. Single Logout

The provider is required to correctly identify the affiliation when responding to a single logout request using the `<LogoutResponse>` message.

- **SP-AFF-006**: If the Principal’s identity federation is between the identity provider and an affiliation group in which the service provider is a member, then the `NameQualifier` attribute MUST contain the unique identifier of the affiliation group. Otherwise, it MUST contain the unique identifier of the service provider. This attribute MUST be used by the providers to identify the specific identity federation of the Principal who is logging out.

2.6.3. Dynamic Proxy of Identity Provider

A service provider may request that an identity provider obtain authentication from a third-party identity provider with which a Principal has already authenticated.

The originator of an authentication request may control proxy behavior by including a `<Scoping>` element where the provider sets a desired `<ProxyCount>` value and/or indicates a list of preferred identity providers which may be proxied by defining an ordered `<IDPList>` of preferred providers.

- **SP-PXY-001**: A service provider MUST be able to construct an `<AuthnRequest>` with valid `<Scoping>`, `<ProxyCount>`, and `<IDPList>` elements.

2.6.4. Name Identifier Mapping

This section describes the requirements for a service provider implementing the Name Identifier Mapping interaction.

- **SP-NIM-001**: The Name Identifier Mapping interaction is a REQUIRED feature of the Extended Service Provider Conformance Profile.

A service provider may play one of two distinct roles in the NameIdentifier Mapping interaction. The first role is the "Requester" which initiates the NameIdentifier request to the identity provider. The second role is the "Target"; the Requester service provider may use the mapped NameIdentifier to issue a `<saml:AttributeQuery>` to the Target to obtain additional information about the Principal.
This interaction MUST NOT be used unless the identity provider metadata element NameIdentifierMappingProtocolProfile specifies the URI http://projectliberty.org/profiles/nim-sp-http.

2.6.4.1. Requester Role

The steps performed by the service provider in the Requester role are illustrated in [LibertyBindProf, Figure 17]. The service provider initiates the NameIdentifier Mapping interaction by sending a SOAP-over-HTTP request to the SOAP endpoint of the identity provider it is querying.

The SOAP message MUST contain exactly one <lib:NameIdentifierMappingRequest> element in the SOAP body, and follow the construction rules defined in [LibertyProtSchema, 3.6.1.1]. In addition, the message MUST be signed.

If the Principal’s identity federation is between the identity provider and an affiliation group in which the Requester service provider is a member, the NameQualifier attribute of the of the request message’s <saml:NameIdentifier> MUST contain the unique identifier of the affiliation group. Otherwise, it MUST contain the unique identifier of the service provider. See [LibertyProtSchema, 3.6.3].

The identity provider will respond to the <lib:NameIdentifierMappingRequest> with a SOAP 200 OK <lib:NameIdentifierMappingResponse> message.

The service provider MUST validate any signature on the response message. The signature on the message MUST be the signature of the <ProviderID> contained in the message. If the signature is not valid, the service provider MUST ignore the message.

The processing rules for the Requester service provider after receiving and validating the <lib:NameIdentifierMappingResponse> are not normatively specified by Liberty. See [LibertyBindProf, 3.7.1.4 and 3.7.1.5] for more information.

2.6.4.2. Target Role

The actions of a service provider in an Target role are not normatively specified by Liberty. See [LibertyBindProf, 3.7.1.4 and 3.7.1.5] for more information.

2.6.4.3. Encrypted Name Identifiers

This section describes the requirements a service provider must meet when decoding and decrypting a NameIdentifier value such as one that might be obtained from an identity provider in the context of [IDP-NIM-006]. Note that in this context, the service provider is playing the role of target (see [role definition]).

This interaction MUST NOT be used unless the provider metadata element NameIdentifierMappingEncryptionProfile specifies the URI urn:liberty:iff:nameid:encrypted.
2.6.4.3.1. Decoding and Decrypting

The processing steps for a provider receiving such an encrypted NameIdentifier are not normatively specified by Liberty, but are described in [LibertyBindProf, 3.8.1.2].

Note that certain block encryption algorithms are mandatory for use with the <xenc:EncryptedData> and therefore must be supported:

- SP-ENI-002 The following algorithms MUST be supported as indicated in [xmlenc-core], sections 5.2.1 and 5.2.2: TRIPLE DES, AES-128, AES-256.

2.6.4.3.2. Security Considerations

- SP-ENI-003 The provider receiving an encrypted NameIdentifier MUST take care to use the IssueInstant andNonce attributes to prevent replay and long-term use of the same encrypted identifier.

2.7. Liberty Enabled Client/Proxy (LECP)

This section contains detailed specifications of the LECP Profile.

2.7.1. General LECP Requirements

- LCP-SSO-001 All HTTP requests made by a LECP MUST include a Liberty-Enabled indication. A Liberty-Enabled indication is either a Liberty-Enabled header or User-Agent header containing a Liberty-Enabled value as defined in [LibertyBindProf, 3.2.4.1].

The preferred Liberty-Enabled indication is the Liberty-Enabled header.

- LCP-SSO-002 A LECP SHOULD add the Liberty-Enabled header to each HTTP request. This header MUST be constructed as specified in [LibertyBindProf, 3.2.4.1].

- LCP-SSO-003 A LECP MAY add a Liberty-Enabled entry in the HTTP User-Agent request header, as specified in [LibertyBindProf, 3.2.4.1]

2.7.2. Single Sign-On

The single sign-on interaction is the only LECP interaction specified. This interaction assumes that the user agent has authenticated at the identity provider and that a valid session exists for the user agent at the identity provider.

- LCP-SSO-004 To initiate single sign-on, the user agent MUST contain at most one Liberty-Enabled header. If a proxy receives a HTTP request that contains a Liberty-Enabled header, it MUST NOT add another Liberty-Enabled header.

- LCP-SSO-005 A proxy MAY replace the Liberty-Enabled header, but this replacement MUST adhere to the specifications in [LibertyBindProf, 3.2.4.2, Step 1]

After receiving the HTTP 200 OK response (containing a <lib:AuthnRequestEnvelope> from the service provider, the LECP will determine the correct identity provider to use.
The LECP MUST issue an HTTP POST of the `<lib:AuthnRequest>` in the body of a SOAP message to the identity provider’s single sign-on service URL. This MUST be the same `<lib:AuthnRequest>` as was received in the service provider’s `<lib:AuthnRequestEnvelope>`. See [LibertyBindProf, 3.2.4.2, Step 4].

In case of any error, the LECP MUST return a `<lib:AuthnResponse>` to the service provider as specified in [LibertyBindProf, 3.2.4.2, Step 4].

After receiving a HTTP response from the identity provider, the LECP will issue an HTTP POST to the service provider.

The HTTP POST from the LECP MUST be composed as specified in [LibertyBindProf, 3.2.4.2, Step 7].

In case of any error, the LECP MUST return a `<lib:AuthnResponse>` to the service provider as specified in [LibertyBindProf, 3.2.4.2, Step 7].
References

Normative


Informative
