SPMainConfig

`shibboleth.xml` contains the majority of configuration information for the Shibboleth SP and is located by default at `/opt/shibboleth-sp/etc/shibboleth/shibboleth.xml`.

```
<SPConfig xmlns="urn:mace:shibboleth:target:config:1.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="urn:mace:shibboleth:target:config:1.0 /opt/shibboleth-sp/share/xml/shibboleth
/shibboleth-targetconfig-1.0.xsd"
          logger="/opt/shibboleth-sp/etc/shibboleth/shibboleth.logger" clockSkew="180">

  <!-- These extensions are "universal", loaded by all Shibboleth-aware processes. -->
  <Extensions>
    <Library path="/opt/shibboleth-sp/libexec/xmlproviders.so" fatal="true"/>
  </Extensions>

  <!-- The Global section pertains to shared Shibboleth processes like the shibd daemon. -->
  <Global logger="/opt/shibboleth-sp/etc/shibboleth/shibd.logger">

    <!--
        <Extensions>
          <Library path="/opt/shibboleth-sp/libexec/shib-mysql-ccache.so" fatal="false"/>
        </Extensions>
        -->

    <!-- Only one listener can be defined. -->
    <UnixListener address="/opt/shibboleth-sp/var/run/shib-shar.sock"/>

    <!-- <TCPListener address="127.0.0.1" port="12345" acl="127.0.0.1"/> -->

    <MemorySessionCache cleanupInterval="300" cacheTimeout="3600" AATimeout="30" AAConnectTimeout="15"
                        defaultLifetime="1800" retryInterval="300" strictValidity="false" propagateErrors="
false"/>
    <!--
        <MySQLSessionCache cleanupInterval="300" cacheTimeout="3600" AATimeout="30" AAConnectTimeout="15"
                           defaultLifetime="1800" retryInterval="300" strictValidity="false" propagateErrors="
false"
                           mysqlTimeout="14400" storeAttributes="false">
          <Argument>&#x2D;&#x2D;language=/opt/shibboleth-sp/share/english</Argument>
          <Argument>&#x2D;&#x2D;datadir=/opt/shibboleth-sp/data</Argument>
        </MySQLSessionCache>
        -->

    <!-- Default replay cache is in-memory. -->
    <!--
        <MySQLReplayCache>
          <Argument>&#x2D;&#x2D;language=/opt/shibboleth-sp/share/english</Argument>
          <Argument>&#x2D;&#x2D;datadir=/opt/shibboleth-sp/data</Argument>
        </MySQLReplayCache>
        -->
  </Global>
  <!-- The Local section pertains to resource-serving processes (often process pools) like web servers. -->
  <Local logger="/opt/shibboleth-sp/etc/shibboleth/native.logger" localRelayState="true">

    <RequestMapProvider type="edu.internet2.middleware.shibboleth.sp.provider.NativeRequestMapProvider">
      <RequestMap applicationId="default">
        <Host name="sp.example.org">
          <Path name="secure" authType="shibboleth" requireSession="true" exportAssertion="true">
            <!-- Example shows a subfolder on the SSL port assigned to a separate <Application> -->
            <Path name="admin" applicationId="foo-admin"/>
          </Path>
        </Host>
      </RequestMap>
    </RequestMapProvider>

    <Implementation>
      <ISAPI normalizeRequest="true">
        <Site id="1" name="sp.example.org">
          <Alias>spalias.example.org</Alias>
```

```
      </Site>
    </ISAPI>
  </Implementation>
</Local>

<!--
         The Applications section is where most of Shibboleth's SAML bits are defined.
         Resource requests are mapped in the Local section into an applicationId that
         points into to this section.
      -->
<Applications id="default" providerId="https://sp.example.org/shibboleth"
              homeURL="https://sp.example.org/index.html"
              xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
              xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">

  <Sessions lifetime="7200" timeout="3600" checkAddress="false"
            handlerURL="/Shibboleth.sso" handlerSSL="false" idpHistory="true" idpHistoryDays="7">

    <!-- This default example directs users to a specific IdP's SSO service. -->
    <SessionInitiator isDefault="true" id="example" Location="/WAYF/idp.example.org"
                      Binding="urn:mace:shibboleth:sp:1.3:SessionInit"
                      wayfURL="https://idp.example.org/shibboleth-idp/SSO"
                      wayfBinding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"/>

    <!-- This example directs users to a specific federation's WAYF service. -->
    <SessionInitiator id="IQ" Location="/WAYF/InQueue"
                      Binding="urn:mace:shibboleth:sp:1.3:SessionInit"
                      wayfURL="https://wayf.internet2.edu/InQueue/WAYF"
                      wayfBinding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"/>

    <!--
        md:AssertionConsumerService elements replace the old shireURL function with an
        explicit handler for particular profiles, such as SAML 1.1 POST or Artifact.
        The isDefault and index attributes are used when sessions are initiated
        to determine how to tell the IdP where and how to return the response.
      -->
    <md:AssertionConsumerService Location="/SAML/POST" isDefault="true" index="1"
                                 Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"/>
    <md:AssertionConsumerService Location="/SAML/Artifact" index="2"
                                 Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01"/>

    <!--
        md:SingleLogoutService elements are mostly a placeholder for 2.0, but a simple
        cookie-clearing option with a ResponseLocation or a return URL parameter is
        supported via the "urn:mace:shibboleth:sp:1.3:Logout" Binding value.
      -->
    <md:SingleLogoutService Location="/Logout" Binding="urn:mace:shibboleth:sp:1.3:Logout"/>

  </Sessions>

  <Errors session="/opt/shibboleth-sp/etc/shibboleth/sessionError.html"
          metadata="/opt/shibboleth-sp/etc/shibboleth/metadataError.html"
          rm="/opt/shibboleth-sp/etc/shibboleth/rmError.html"
          access="/opt/shibboleth-sp/etc/shibboleth/accessError.html"
          supportContact="root@localhost"
          logoLocation="/shibboleth-sp/logo.jpg"
          styleSheet="/shibboleth-sp/main.css"/>

  <!-- Indicates what credentials to use when communicating -->
  <CredentialUse TLS="defcreds" Signing="defcreds">
    <!-- RelyingParty elements can customize credentials for specific IdPs/sets. -->
    <!--
        <RelyingParty Name="urn:mace:inqueue" TLS="inqueuecreds" Signing="inqueuecreds"/>
        -->
  </CredentialUse>

  <!-- Use designators to request specific attributes or none to ask for all -->
  <!--
      <saml:AttributeDesignator AttributeName="urn:mace:dir:attribute-def:eduPersonScopedAffiliation"
                                AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri"/>
      -->
```

```
    <!-- AAP can be inline or in a separate file -->
    <AAPProvider type="edu.internet2.middleware.shibboleth.aap.provider.XMLAAP" uri="/opt/shibboleth-sp/etc
/shibboleth/AAP.xml"/>

    <!-- Dummy metadata for private testing, delete for production deployments. -->
    <MetadataProvider type="edu.internet2.middleware.shibboleth.metadata.provider.XMLMetadata"
                      uri="/opt/shibboleth-sp/etc/shibboleth/example-metadata.xml"/>

    <!-- InQueue pilot federation, delete for production deployments. -->
    <MetadataProvider type="edu.internet2.middleware.shibboleth.metadata.provider.XMLMetadata"
                      uri="/opt/shibboleth-sp/etc/shibboleth/IQ-metadata.xml"/>

    <!-- The standard trust provider supports SAMLv2 metadata with path validation extensions. -->
    <TrustProvider type="edu.internet2.middleware.shibboleth.common.provider.ShibbolethTrust"/>

    <saml:Audience>urn:mace:inqueue</saml:Audience>

    <!--
          <Application id="foo-admin">
            <Sessions lifetime="7200" timeout="3600" checkAddress="true"
                      handlerURL="/secure/admin/Shibboleth.sso" handlerSSL="true"
                      cookieProps="; path=/secure/admin; secure"/>
            <saml:AttributeDesignator AttributeName="urn:mace:dir:attribute-def:eduPersonPrincipalName"
                                      AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri"/>
          </Application>
          -->

  </Applications>

  <!-- Define all the private keys and certificates here that you reference from <CredentialUse>. -->
  <CredentialsProvider type="edu.internet2.middleware.shibboleth.common.Credentials">
    <Credentials xmlns="urn:mace:shibboleth:credentials:1.0">
      <FileResolver Id="defcreds">
        <Key>
          <Path>/opt/shibboleth-sp/etc/shibboleth/sp-example.key</Path>
        </Key>
        <Certificate>
          <Path>/opt/shibboleth-sp/etc/shibboleth/sp-example.crt</Path>
        </Certificate>
      </FileResolver>

      <!--
          Mostly you can define a single keypair above, but you can define and name a second
          keypair to be used only in specific cases and then specify when to use it inside a
          <CredentialUse> element.
            -->
      <!--
          <FileResolver Id="inqueuecreds">
            <Key>
              <Path>/opt/shibboleth-sp/etc/shibboleth/inqueue.key</Path>
            </Key>
            <Certificate>
              <Path>/opt/shibboleth-sp/etc/shibboleth/inqueue.crt</Path>
            </Certificate>
          </FileResolver>
          -->
    </Credentials>
  </CredentialsProvider>

  <!-- Specialized attribute handling for cases with complex syntax. -->
  <AttributeFactory AttributeName="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
                    type="edu.internet2.middleware.shibboleth.common.provider.TargetedIDFactory"/>

</SPConfig>
```

Main Configuration Elements

These container elements specify broad functionality for the Shibboleth SP.

<SPConfig clockSkew="*integer*">	This is the root element for Service Provider configuration and must be present once and only once. It must always contain a `Global` element, a `Local` element, an `Applications` element, one or more `CredentialsProvider` elements, and optionally an `Extensions` element. `clockSkew` defines allowed clock skew in seconds between SP and !IdP servers when evaluating times sent in messages. Defaults to 180, and should be as small as practical.

<Global logger="*pathname*">	This is the container element for configuration information pertaining to `shibd` and other shared Shibboleth processes. Its single attribute, `logger` , points to a Log4CPP property configuration file that controls `shibd` logging behavior. It is placed within the `SPConfig` element and may contain several other elements, including an `Extensions` element specifying additional libraries. It must contain either a `UnixListener` element to listen to the server module on a UNIX domain socket or a `TCPListener` element to listen on a TCP port. Session caching must also be specified using a `MemorySessionCache` element to use in-memory session caching or a `MySQLSessionCache` element to backup session information into a MySQL database.

<Local logger="*pathname*">	This is the container element for configuration information pertaining to the integration of the service provider into the web server environment. Its single attribute, `logger` , points to a Log4CPP property configuration file that controls logging behavior. It is placed within the `SPConfig` element and may contain an `Extensions` element specifying additional libraries. It may also contain an `Implementation` element, describing additional session handling mechanics for other web environments.

It must contain a `RequestMapProvider` element, which provides fine-grained control over aspects of SP behavior at a host, path, or document level.

Listeners

There are three different kinds of listening performed by components of the Shibboleth SP. The `TCPListener` and `UnixListener` provide internal communication between pieces of the SP, while the `AssertionConsumerService` elements define how and where the SP as a whole responds to requests from other providers. Finally, the SP as a whole helps broker access to web applications, and for some purposes in the Shibboleth flows must redirect the web browser to other sites; destinations for these redirects are defined on a per-application basis using SessionInitiator elements.

<TCPListener address="*pathname*" port="*integer*" acl="*ip*"/>	This element is placed within the `Global` element and is mutually exclusive with the `UnixListener` and `Listener` elements. It allows `shibd` to communicate with the web server component using TCP.

- `address` : Specifies the IP address of the listener.
- `port` : Specifies the TCP port on which `shibd` will listen to requests to obtain attributes.
- `acl` : By default, the `shibd` will only listen to requests from 127.0.0.1 (localhost). This should generally not be specified as anything else except in test environments for security purposes.

<UnixListener address="*pathname*"/>	Use this element to specify a UNIX domain socket located at the `pathname` specified in the `address` attribute at which `shibd` should listen for requests. This element must be contained by the `Global` element and is mutually exclusive with the `TCPListener` and `Listener` elements. `UnixListener` cannot be specified for Windows-based installations.

One or more `SessionInitiator` elements can be defined to help streamline IdP selection and the WAYF functionality. A new `SessionInitiator` is appropriate when an application is accessed by a different community than other applications.

<SessionInitiator isDefault="*true/false*" id="*uniqueID*" Location="*pathname*" Binding="urn:mace:shibboleth:sp:1.3:SessionInit" wayfURL="*URL*" wayfBinding="urn:mace:shibboleth:1.0:profiles:AuthnRequest" />	This element is placed within the `Sessions` element to define one way a Shibboleth session initiation process can be triggered. Only one `<SessionInitiator>` may be marked default. The `id` is used for references within `shibboleth.xml` , while the `location` is used to define a URL for this handler to be used in lazy session initiation. The `wayfURL` is the location the browser will be redirected to for user authentication to eventually occur, and may point at a WAYF-style redirection engine or directly at an !IdP's SSO handler.

Session Caching

Shibboleth maintains several sessions between users and SP's in addition to sessions maintained at the IdP and by applications themselves. Inappropriate timeout settings can either lead to service interruptions or security vulnerabilities

<MemorySessionCache AAConnectTimeout="*seconds*" AATimeout="*seconds*" cacheTimeout="*seconds*" cleanupInterval="*seconds*" defaultLifetime="*seconds*" propagateErrors="*true/false*" retryInterval="*seconds*" strictValidity="*true/false*"/>	Shibboleth will cache sessions and received attributes in memory if this element is found in the `Global` element. This element is mutually exclusive with the `MySQLSessionCache` and `SessionCache` elements.

- `AAConnectTimeout` : Time in seconds the SP will wait before timing out on the initial connection to an IdP to request attributes. Defaults to `15` .
- `AATimeout` : Time in seconds the SP will wait before timing out while waiting for attributes from an IdP once the initial connection is established. Defaults to `30` .
- `cacheTimeout` : Time in seconds to permit a session to stay in the cache before being purged. Defaults to `28800` .
- `cleanupInterval` : Seconds between runs of the background thread that purges expired sessions. Defaults to `300` .
- `defaultLifetime` : If the attribute assertion doesn't carry an explicit expiration time, the assertion will expire after this time in `seconds` has elapsed. Defaults to `1800` .
- `propagateErrors` : If true, then any errors that occur during the attribute query stage are fatal and will be presented to the user as an error, terminating their session. If false, any errors that occur during the query are non-fatal, and the application will be given older, expired attributes based on the `strictValidity` setting.
  This should generally only be left to false (the default) by deployments that are using real principal names as subjects because attribute retrieval is treated as an optional process.

- `retryInterval` : Time in seconds between attempts to obtain fresh attributes. If a query fails, a timer is set, and once the interval elapses, the next user request causes another query. This prevents pointless repeated attempts to query a failed !IdP. Defaults to `300` .
- `strictValidity` : If true, expired attributes will never be made available to the Shibboleth application; if no valid attributes can be obtained, then an empty set is provided. When false, if a fresh set of attributes cannot be retrieved due to failures, any cached, expired attributes are made available. Defaults to `true` .

<MySQLSessionCache mysqlTimeout="*seconds*"/>	Shibboleth will back the memory cache of sessions using an embedded MySQL database if this element is found in the `Global` element. Arguments may be passed directly to MySQL by populating this element with `Argument` elements. The element may also specify any of the attributes defined for the `MemorySessionCache` element. Mutually exclusive with the `MemorySessionCache` and `SessionCache` elements. This is most useful for deploying the SP in balanced environments where sessions must be tracked across multiple web servers.

- `mysqlTimeout` : Time in seconds to permit a session to stay in the persistent cache before being purged. Defaults to `28800` .

<SessionCache type="*string*"/>	Specifies a pluggable session cache implementation of the specified `type` . This element is placed within the `Global` element and is mutually exclusive with the `MemorySessionCache` and `MySQLSessionCache` elements. Any plugin should support the basic attributes defined by the `MemorySessionCache` element.

%COMMENT%

- It might be worth mentioning that the SessionInitiator element can also be used in flows initiated at an IdP or portal site, by including the "providerId" parameter in the request. This is documented in the default 1.3 shibboleth.xml file. It isn't really a lazy session flow, as the application doesn't need to be aware of it. -- Main.IanYoung - 24 May 2006 14:11:49