# Azure Active Directory Single Sign-On with Shibboleth Identity Provider 3

## Table of Contents

[Azure Active Directory Single Sign-On with Shibboleth Identity Provider 3 1](#_Toc516087363)

[Table of Contents ii](#_Toc516087364)

[Introduction to this technical article 3](#_Toc516087365)

[Installing the Shibboleth IdP V3 software 3](#_Toc516087366)

[Configuring the Shibboleth IdP 7](#_Toc516087367)

[Configuring Shibboleth for use with single sign-on 12](#_Toc516087368)

[Installing Azure Active Directory Module for PowerShell (MSOnline) 19](#_Toc516087369)

[Connecting PowerShell to Azure AD 20](#_Toc516087370)

[Setting up a trust between Shibboleth IdP and Azure AD 20](#_Toc516087371)

[Provision user principals to Azure AD 25](#_Toc516087372)

[Verifying single sign-on with Shibboleth IdP 26](#_Toc516087373)

[Troubleshooting the single sign-on (SSO) with Shibboleth IdP 28](#_Toc516087374)

[Understanding how federated authentication works in Office 365 29](#_Toc516087375)

[Understanding the Shibboleth IdP configuration (REVISION OF THIS SECTION REQUESTED) 29](#_Toc516087376)

[Understanding the authentication flows (REVISION OF THIS SECTION IS UNDERWAY) 36](#_Toc516087377)

[Appendix A. Glossary of Shibboleth Terms 41](#_Toc516087378)

[Appendix B. Sample Shibboleth IdP v3 configuration files 43](#_Toc516087379)

[attribute-resolver.xml sample 43](#_Toc516087380)

[attribute-filter.xml sample 44](#_Toc516087381)

[metadata-provider.xml sample 44](#_Toc516087382)

[relying-party.xml sample 44](#_Toc516087383)

[saml-nameid.xml sample 45](#_Toc516087384)

## Introduction to this technical article

Through its support for the SAML 2.0 protocol, Shibboleth Consortium’s Identity Provider (IdP) v3 provides claims-based web single sign-on with Azure Active Directory (Azure AD) and Azure AD protected resources, such as Microsoft Azure, Office 365, and its web and e-mail rich client applications like Outlook.

Building on existing documentation, this document is intended to provide a better understanding of the different single sign-on deployment options for Azure AD, and to describe how to enable single sign-on using corporate credentials and the Shibboleth IdP v3 to Azure AD, and the different configuration elements to be aware of for such deployment.

This document is intended for system architects and IT professionals who are interested in understanding the basics of the single sign-on feature of Azure AD with Shibboleth IdP v3 along with planning and deploying such a system in their environment.

## Installing the Shibboleth IdP V3 software

Administrators on a Windows Server platform can use the Windows .msi installation package as an alternative to the .zip archive available on the Shibboleth Community web site.

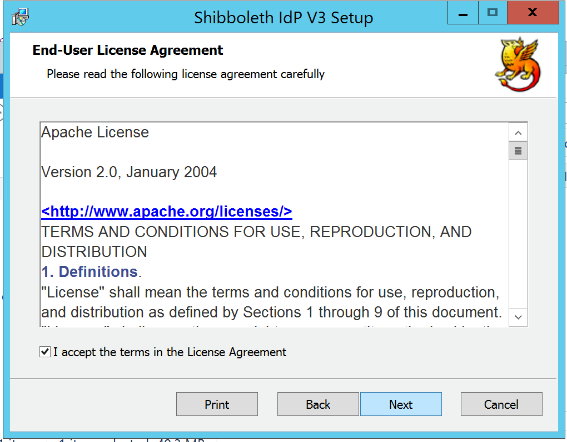
This enables to save a considerable amount of time, effort, and work at configuration time, as long as you enter the required information accurately at installation time. (Eclipse Jetty is included with the Windows installer, and Apache httpd is not required.)

To install the Shibboleth IdP software package onto the IDP0 (*idp0.shib.idmgt.contoso.com*) machine, proceed with the following steps:

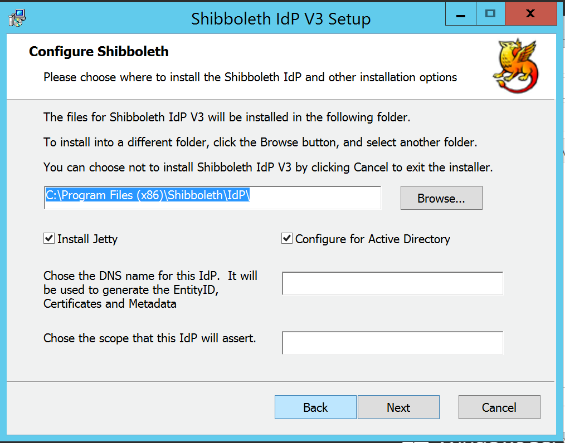
1. Visit the [Shibboleth IdP download site](http://shibboleth.net/downloads/identity-provider/latest/)[[1]](#footnote-1) and download the latest Shibboleth Identity Provider software package. This guide uses the following Windows .msi installation package: shibboleth-identity-provider-3.3.2.0 -x64.msi.
2. Click **Run** to execute shibboleth-identity-provider-3.3.2.0-x64.msi. The Shibboleth IdP V3 Setup wizard opens up.



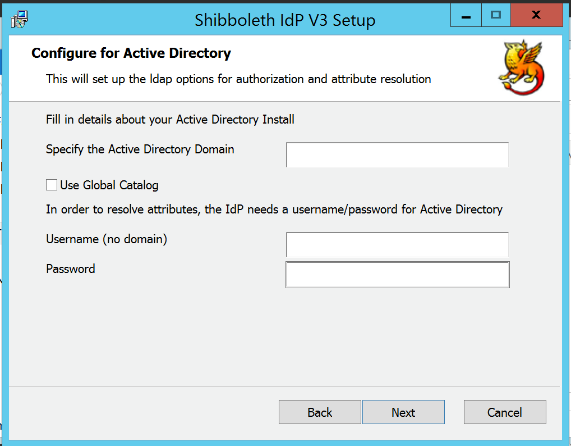
1. In the **Welcome** page, click **Next**.



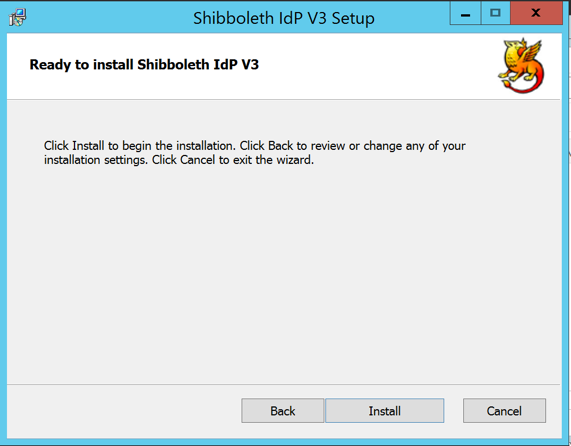
1. In the **License Agreement** page, click **Next**.



1. For this page:
   1. In the destination folder page, keep the default path.
   2. Select **Install Jetty**. This will install the Jetty servlet container.
   3. Select **Configure for Active Directory**. This will present the Active Directory (LDAP) configuration options page.
   4. Type “*idp0.shib.idmgt.contoso.com*” in **Choose the DNS name of the IdP.** This value is the public-facing hostname of the IdP and is used to name the Shibboleth IdP endpoints in the generated metadata and back channel certificate.
   5. Leave empty the field **Choose the scope that this IdP will assert?** In most (or federated) deployments, this value would be the domain that this IdP represents (e.g. *example.org*, *idmgt.contoso.com*).
   6. Click **Next**.



1. In such case, this page and the next one have no suggested entries.
2. Type “*localhost*” in **Specify the Active Directory Domain**. This is the hostname of the LDAP domain component from which the Shibboleth IdP will serve authentication and attributes.
3. Leave the **Use Global Catalog** unchecked.
4. Type “CN=ShibSvc,CN=Users,DC=SHIB,DC=IDMGT,DC=CONTOSO,DC=COM” in **Username (no domain)**, and “Password1” in **Password**.
5. Click **Next**.



1. Click **Next**. The wizard proceeds with the installation.



1. Click **Finish**.

**Note** This installer automatically installs and configures [Eclipse Jetty](http://www.eclipse.org/jetty/)[[2]](#footnote-2). Jetty depends on an existing 32-bit or 64-bit Java JRE installation.

*The C:\Program Files (x86)\Shibboleth\IdP* directory should now contain the following folders:

* ***bin****:* contains command line tools.
* ***conf***: contains the IdP’s configuration files.
* ***credentials***: contains the credentials used by the IdP. The Shibboleth IdP installation process generates a set of three long-life self-signed certificate and private keys which are saved in this folder: *idp-backchannel.crt*, *idp-backchannel.key*, *idp-encryption.crt*, *idp-encryption.key*, and *idp-signing.crt*, *idp-signing.key*. Also, stored here are the *sealer.kver* and *sealer.jks* which are used to encrypt the contents of cookies used to maintain and SSO session.
* ***dist****:* contains a copy of the baseline configuration files.
* ***doc****:* contains license files and text files.
* ***edit-webapp****:* contains files that are new or will override the baseline equivalent web components with local customizations.
* ***flows****:* contains user definable or modifiable flows.
* ***jetty-base****:* contains IdP specific files used by the built-in Jetty servlet container. Referred to as JETTY\_BASE.
* ***logs***: is the location of the Shibboleth log files:
  + *idp-access.log*: record of all the clients that connect to the IdP;
  + *idp-audit.log*: record of all information sent out from the IdP;
  + *idp-consent-audit.log*: record of all user consent actions;
  + *idp-process.log*: detailed description the IdP processing requests.
* ***messages****:* contains a *messages.properties* files that store overrides to the baseline literal text string presented by the IdP and error handler messages and mappings.
* ***metadata***: is the default location where various metadata files are stored. The IdP does not automatically load any metadata. Metadata read from a file, or stored backup copies of remote metadata are usually put in this directory.
* ***static****:* an empty directory that is used by Jetty to map requests for static resource to.
* ***system****:* contains files that should not be modified by end-users. These files maybe over-written by future Shibboleth upgrades.
* ***views****:* contains Velocity Templates that are used when rending IdP web pages.
* ***war***: is the location of the IdP WAR file created by the installer. A WAR file (or web application ARchive) is a JAR file used to distribute a collection of JavaServer Pages, Java Servlets, Java classes, XML files, and other resources that together constitute a Java web application such as the Shibboleth IdP.
* ***webapp****:* contains baseline web application files that should not be modified by the end-user. If changes are necessary, the desired file should be copied over to the *edit-webapp* directory.

*The C:\Program Files (x86)\Shibboleth\IdP\bin* directory should now contains the *build.bat* batch file that enables you to regenerate the WAR file for any other reason. The “captive” Jetty configuration is pointed to this file, instead of copying it to its folders, so that new WARs are automatically taken into account if you rebuild the IdP (to add an extension, for example) or run into problems with Jetty file caching mechanisms.

The Shibboleth documentation refers to this directory as IDP\_HOME.

Likewise, the *C:\Program Files (x86)\Shibboleth\Jetty* directory contains the “captive” Jetty. The Shibboleth IdP documentation refers to this directory as JETTY\_HOME.

## Configuring the Shibboleth IdP

The Shibboleth IdP software must be configured once installed. The configuration typically requires:

* Configure the SSL/TLS certificate for the “captive” Jetty environment,
* Configure the users authentication for the IdP,
* Define the source of user attributes as well as the mapping between attributes names,
* Define which attributes to release to which service provider such as Azure AD in our context.

The last bullets relate to the attributes to release in SAML 2.0 assertions for a service provider (SP). The Shibboleth IdP software is preconfigured to include several attributes in the SAML 2.0 assertions it generates, including an example of eduPersonScopedAffiliation and eduPersonTargetedID. For the moment, we do not modify this part.

The configuration part that corresponds to these two bullets will be instead specifically covered in section § Configuring Shibboleth for use with single sign-on where we will add Azure AD as a relying party and modify various Shibboleth configuration files.

For the rest, the following sections describe what files are to be edited to configure the IdP. The paths used reflect our test installation as depicted above and should be changed to reflect your own configuration.

**Note** For information on the configuration of the Shibboleth IdP software, see the online help topic [Configuration](https://wiki.shibboleth.net/confluence/display/IDP30/Configuration)[[3]](#footnote-3) on the Shibboleth Community wiki.

The Shibboleth IdP uses the following primary configuration files to control various aspects of its operation:

|  |  |
| --- | --- |
| **Configuration file** | **Description** |
| attribute-filter.xml | Configures the release of attributes to SPs. |
| attribute-resolver.xml | Configures attribute collection, transformation, and encoding from source systems. |
| metadata-provider.xml | Configures the SPs that are trusted by the IdP. |
| relying-party.xml | Configures how the IdP processes messages that are received and sent. |
| logback.xml | Configuration of the IdP's logging system. You might want to use this to debug problems. |
| saml-nameid.properties saml-nameid.xml | Configuration for Subject and subject NameId formats. |
| idp.properties | General IdP configuration file. |
| ldap.properties | LDAP configuration for the Username/Password authentication mechanism for forms based authentication and attribute resolver. |

These configuration files are at %IDP\_HOME%\conf (i.e. *C:\Program Files (x86)\Shibboleth\IdP)*.

**Note** In the following configuration files excerpts, comments may be omitted.

### Configuring the SSL/TLS certificate for the Jetty

In section § Requesting a SSL/TLS certificate, you’ve acquired a SSL/TLS certificate as a .pfx file (idp0.shib.idmgt.contoso.com.pfx).

The “captive” Jetty configuration in the *%IDP\_HOME%\jetty-base\start.d\idp.ini* file points to a .p12 created by the Shibboelth IdP installer. To use our key and certificate file, we can either rename our file to ***idp-userfacing.p12*** and place it in *%IDP\_HOME%\credentials* directory, or we can update the *idp.ini*’s **jetty.browser.keystore.path** property to point to the .pfx file.

Update the **jetty.browser.keystore.password** property to the password set on the .pfx file.

### Configuring Shibboleth users authentication

For the purposes of this document, we’ll let the Shibboleth IdP authenticate users via their username/password credentials against our organizational LDAP.

The Shibboleth IdP installer usually sets up the *IDP\_HOME\conf\ldap.properties* appropriately, but it is worth reviewing the key settings to ensure accuracy.

|  |  |
| --- | --- |
| **Configuration file** | **Description** |
| idp.authn.LDAP.authenticator | Indicates the configuration type. (should be set to *adAuthenticator*.) |
| idp.authn.LDAP.ldapURL | *LDAP Connection URL string. (should be ldap://localhost:389)* |
| idp.authn.LDAP.baseDN | The base of the LDAP tree to search for user accounts. (Should be **CN=Users,DC=SHIB,DC=IDMGT,DC=CONTOSO,DC=COM** ) |
| idp.authn.LDAP.userFilter | The filter to search LDAP with. “{user}” will be substituted for the username captured in the login page. (Should be **(sAMAccountName={user})** ) |
| idp.authn.LDAP.bindDN | Search account distinguished name (or UPN) (should be **CN=ShibSvc,CN=Users,DC=SHIB,DC=IDMGT,DC=CONTOSO,DC=COM**) |
| idp.authn.LDAP.bindDNCredential | Search account password. (should be **Password1**) |
| idp.authn.LDAP.dnFormat | The UPN format. (should be **%s@shib.idmgt.contoso.com**) |
| idp.attribute.resolver.LDAP.searchFilter | The filter used for attribute lookup. (should be **(sAMAccountName=$resolutionContext.principal)** ) |
| idp.attribute.resolver.LDAP.returnAttributes | The user attributes to retrieve for attribute resolution. (should be **mail** and **uid**) |

**Note** Replace “Password1” by your own password previously set in section § Creating a Shibboleth Service account for the Shibboleth service user.

**Note** For more information, see the online help topic [LDAPAuthnConfiguration](https://wiki.shibboleth.net/confluence/display/IDP30/LDAPAuthnConfiguration)[[4]](#footnote-4) on the Shibboleth Community wiki.

### Adjusting the level of logging

At the top of the *%IDP\_HOME%\conf\****logback.xml*** file, are three loggers defined for Shibboleth, SAML and LDAP messages, and the PROTOCOL\_MESSAGE logger in comments. When you are just starting out, or trying to resolve a problem, it is a good idea to change the log level to DEBUG in all of these, and remove the comments from around the PROTOCOL\_MESSAGE logger. Specifying DEBUG causes the log file produced to be more comprehensive and informative, but much larger, so you should turn the log level to INFO or WARN once you are happy with the configuration or the problem is resolved.

The Shibboleth log files are written to the logs subdirectory of the Shibboleth installation directory; the *idp-process.log* is usually the most informative.

The following settings in *%IDP\_HOME%\conf\idp.properties will override the logback.xml* and may give the right level of information to start debugging:

idp.loglevel.idp=-DEBUG

idp.loglevel.opensaml=-DEBUG

idp.loglevel.ldap=-DEBUG

idp.loglevel.messages=-DEBUG

**Note** For more information, see the online help topic [[Logging](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPLogging)Configuration](https://wiki.shibboleth.net/confluence/display/IDP30/LoggingConfiguration)[[5]](#footnote-5) on the Shibboleth Community wiki.

Removing the properties from the *idp.properties* file will restore the default logging levels.

As with any change to IdP configuration files, you need to restart the Java servlet container (e.g. the “captive” Jetty) or the IdP application for it to pick up the changes.

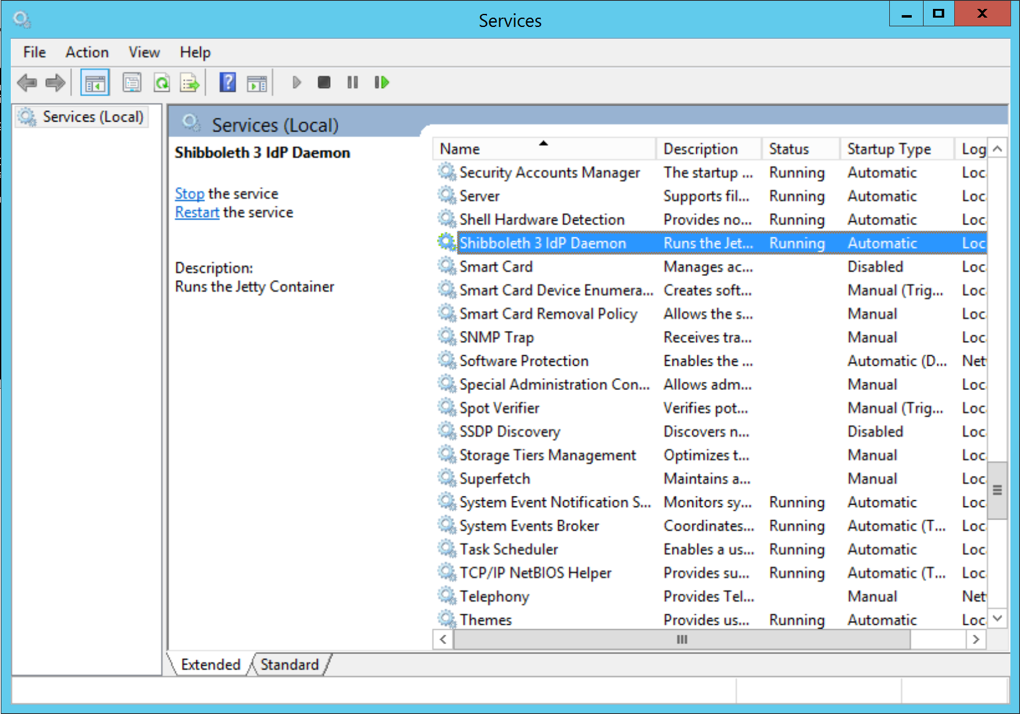
Proceed with the following steps:

1. Use Windows Explorer to navigate to %IDP\_Home%\conf, e.g. C:\Program Files (x86)\Shibboleth\IdP\conf.
2. Right-click the *idp.properties* file, and then click **Edit**. The file should open in Notepad.
3. Set the **logger** properties to reflect the above configuration.
4. Save and close the *idp.properties* file.

### Restarting the Shibboleth web server to take into account the updated configuration

To restart the Shibboleth web server and check for start-up errors, proceed with the following steps:

1. Open the Windows Service Manager. Select Shibboleth 3 IdP Daemon entry.



1. Click **Stop/Square** button in the toolbar.
2. Use Windows Explorer to navigate to %IDP\_Home%\logs, e.g. C:\Program Files (x86)\Shibboleth\IdP\logs.
3. Delete all existing logs.
4. On the **Service Manager** interface, click **Start/Play** near the bottom.
5. Check the files in *%IDP\_Home%\logs* for errors and search/troubleshoot as necessary. If you are still stumped, please check out [troubleshooting](https://wiki.shibboleth.net/confluence/display/IDP30/Troubleshooting)[[6]](#footnote-6) on the Shibboleth Community wiki.

The above instructions assume you are using Captive Jetty, which should normally be the case if you’ve followed the previous instructions.

### Ensuring that the Shibboleth IdP is running

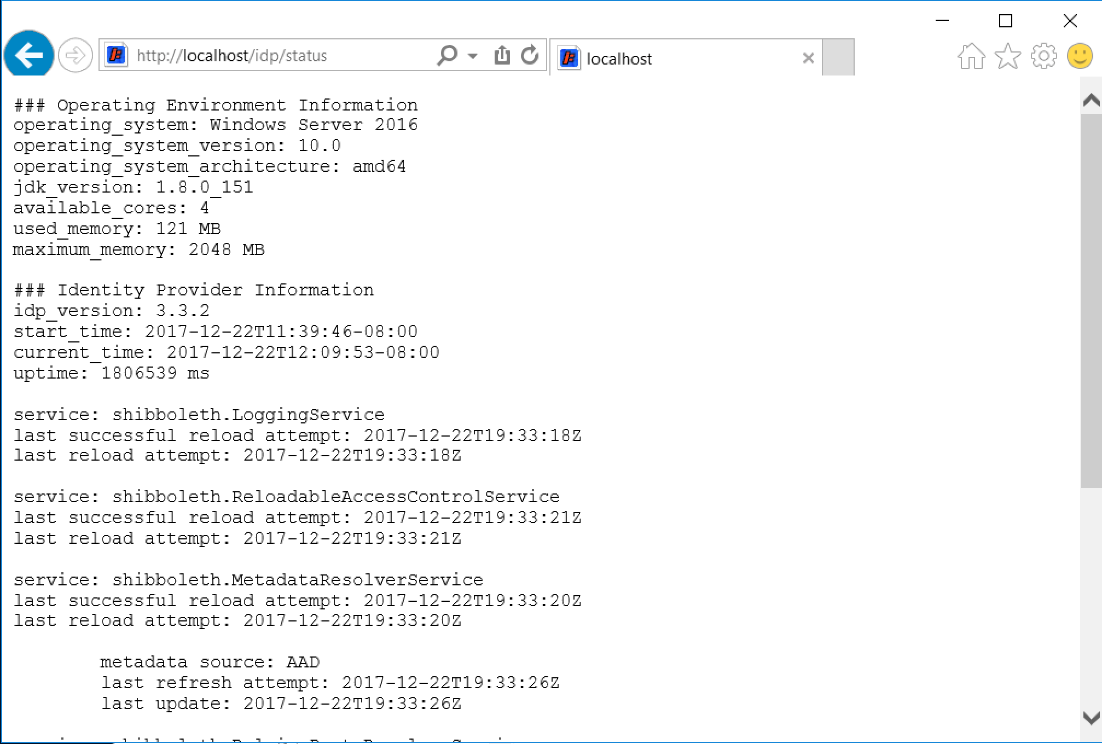
To sign in to Tomcat Manager to ensure that Shibboleth IdP is running, proceed with the following steps:

1. Launch Internet Explorer and browse to [https://idp0.shib.idmgt.contoso.com](https://idp0.shib.idmgt.archims.fr)/idp/
2. If a Shibboleth branding IdP does not appear, you likely have an error in *%IDP\_Home%\logs*. If you are still stumped, please check out [Troubleshooting](https://wiki.shibboleth.net/confluence/display/IDP30/Troubleshooting)[[7]](#footnote-7) at the Shibboleth Community wiki site.

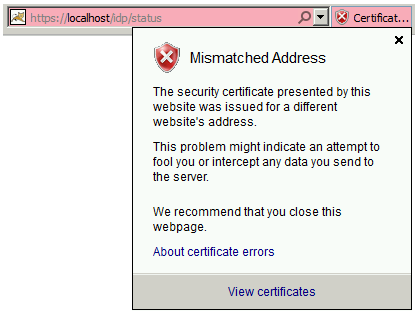
### Performing configuration tests

Proceed with the following steps:

1. Use Windows Explorer to navigate to %IDP\_Home%\conf, e.g. C:\Program Files (x86)\Shibboleth\IdP\conf.
2. Attempt to visit the link for the Federation Metadata: <https://idp0.shib.idmgt.contoso.com/idp/shibboleth>
3. Attempt to visit the link for the IdP Status Page (<https://localhost/idp/status>). You should see a page like the following one. If this is the case, you can continue with the next section.



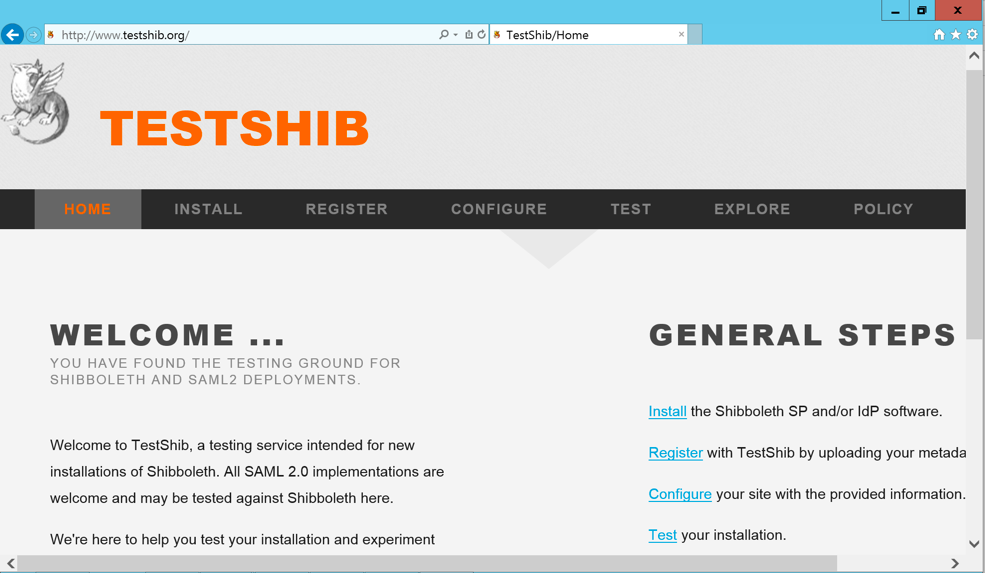
Please note that the Address Bar turns red to signify that the page is protected by an SSL/TLS certificate that is issued for a different website’s address: *idp0.shib.idmgt.contoso.com*.



You can optionally use the TestShib service to get to the stage where you can log in to the IdP and authenticate against the organization's LDAP based on AD LDS.

To (optionally) further test your Shibboleth IdP installation, you can use the TestShib service as follows:

1. Navigate to the TestShib service at <http://www.testshib.org/index.html>.



1. Click **Register** to create a new IdP and follow the instructions.

In a production environment, all references to the TestShib service should be removed from the IdP configuration files.

## Configuring Shibboleth for use with single sign-on

This section contains guidelines on how to configure Shibboleth Identity Provider (IdP) software to be used with Azure AD to enable single sign-on access to one or more Microsoft cloud services (such as Office 365 or Microsoft Azure) using the SAML 2.0 protocol. The SAML 2.0 relying party for a Microsoft cloud service used in this scenario is Azure AD. It describes what files are to be edited to appropriately configure the Shibboleth IdP. The paths used reflect our test installation and should potentially be changed to reflect your configuration.

In the rest of this document topic, the environment variable IDP\_HOME is the base directory where you installed Shibboleth IdP, for example in our case, *C:\Program Files (x86)\Shibboleth\IdP*. Be sure to replace IDP\_HOME with your own specific path.

**Note** For information, see the Microsoft TechNet article [Configure Shibboleth for use with single sign-on](http://technet.microsoft.com/en-us/library/jj205463)[[8]](#footnote-8).

### Adding Azure AD metadata

Adding a partner like Azure AD into Shibboleth IdP consists in defining it in the %*IDP\_Home%\conf\metadata-provider****.xml*** file. Generally speaking, this file defines how the Shibboleth IdP should interact with service providers in the federation and how it gets the federation metadata via the definition of a metadata provider.

The partner definition simply consists of referencing the partner’s XML metadata document via a new metadata provider (*<MetadataProvider>* element).

Shibboleth IdP indeed needs information about the Azure AD relying party. Azure AD publishes metadata at <https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml>.

The following two metadata provider definitions enable to add the above metadata to the Shibboleth IdP:

1. The file system metadata provider: Manually download and store Azure AD metadata in a file in the *IDP\_HOME/metadata* folder.

-or-

1. The file backed HTTP metadata provider: configure Shibboleth IdP to pull the Azure AD metadata directly.

We preferentially use the latter below option.

**Note** Each type of metadata provider has its own set of configuration options. For information on the metadata provider, see the online help topic [MetadataConfiguration](https://wiki.shibboleth.net/confluence/display/IDP30/MetadataConfiguration)[[9]](#footnote-9) on the Shibboleth Community wiki site.

**Note** Azure AD does not read metadata from the identity provider.

1. You also have to specify the Shibboleth IdP where to find the Azure AD metadata document. As previously discussed, you can do this by adding another entry to the *metadata-provider.xml* file. Still in Notepad with the *metadata-provider.xml* file opened, Press Ctrl+F to find “*</MetadataProvider>*”.
2. Move to the next line down and insert the following text to use the file backed HTTP metadata provider:

<!—- Microsoft Azure AD Metadata -->

<MetadataProvider id="AAD" xsi:type="FileBackedHTTPMetadataProvider"

metadataURL="https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml"

backingFile="C:\Program Files (x86)\Shibboleth\IdP\metadata\AAD-FederationMetadata.xml" />

The file backed HTTP metadata provider loads the Azure AD metadata XML file via HTTP and backs it up to a local file, for example in our case, *%IDP\_HOME\metadata\AAD-FederationMetadata.xml*.

Depending on your network configuration, you may require to interact with an HTTP proxy. In such case, additional parameters are required in the MetadataProvider element such as *proxyHost* and *proxyPort*. Please refer to the Shibboleth documentation for additional information.

If you rather want to use the file system metadata provider, the related declaration is as follows:

<!—- Microsoft Azure AD Metadata -->

<MetadataProvider id="AAD" xsi:type="FilesystemMetadataProvider"

metadataFile=”C:\Program Files (x86)\Shibboleth\IdP\metadata\AAD-FederationMetadata.xml"/>

1. Save and close the ***metadata-provider.xml*** file.

### Configuring a relying party override

Azure AD does not process authentication responses that are encryped, which the Shibboleth IdP does by default.

To configure Azure AD relying party override, proceed with the following steps:

1. Use Windows Explorer to navigate to %IDP\_Home%\conf, e.g. C:\Program Files (x86)\Shibboleth\IdP\conf.
2. Right-click the ***relying-party.xml*** file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “shibboleth.RelyingPartyOverrides”.
4. Move to the next line down and insert the following text.

<util:list id="shibboleth.RelyingPartyOverrides">

<!—- Microsoft Azure AD -->

<bean parent="RelyingPartyByName" c:relyingPartyIds="urn:federation:MicrosoftOnline">

<property name="profileConfigurations">

<list>

<bean parent="SAML2.SSO" p:encryptAssertions="false" />

</list>

</property>

</bean>

Make sure that:

* The relying party*id* value matches the *entityID* value of the *EntityDescriptor* element of the Azure AD metadata, for example as of today "*urn:federation:MicrosoftOnline*";

### Configuring the Shibboleth IdP attribute resolver

Azure AD requires two pieces of data from Shibboleth IdP to locate the shadow account in the authentication platform.

1. **Azure AD ImmutableID.** Azure AD requires that you select a unique identifier for each user in your user directory. You must also configure Shibboleth IdP to send this attribute on each federated login to Azure AD in the SAML 2.0 *NameID* assertion. This identifier must not change for this user over the lifetime of the user being in your system. Azure AD calls this attribute the *ImmutableID*. The value for the unique identifier must not contain domain information and is case-sensitive. For example, do not use *user@shib.idmgt.contoso.com*.

The value used here with AD LDS will be (as an illustration) the *uid* property that we’ve previously provisioned for the test user user1 (see section § Creating and configure a test user for the Shibboleth IdP). It can also be for example the Shibboleth eduPersonTargetedID attribute that is calculated by default.

When creating accounts, you must ensure the ImmutableID is processed the same way, otherwise the user will not be able to login to the Microsoft Cloud services. For instance, with Active Directory, the Azure AD Connect tool automatically uses the Active Directory objectGUID for the ImmutableID value and processes the ImmutableID the same way. We mimic here the approach.

1. **Azure AD UserID.** Azure AD requires that the Azure AD User ID, for example, *user@shib.idmgt.contoso.com*, is sent. With Active Directory, the value is stored in the LDAP ***UserPrincipalName*** attribute. With AD LDS, we will use instead the mail attribute that we’ve provisioned earlier with the user e-mail address.

As outlined before, the Shibboleth IdP can retrieve attributes from Active Directory, another LDAP directory, a SQL database, etc., generate attributes based on other attributes, or define them statically.

For that purpose, the %*IDP\_Home%\conf\****attribute-resolver.xml*** file defines how the IdP generates SAML 2.0 attributes for the IdP's users. It specifies how to configure the IdP to authenticate users against the organization's attributes source(s), e.g. AD LDS in our configuration, how to use it to look up values associated with those users, and how to use these as the basis for attribute generation.

The file more particularly defines:

1. Data connectors (*<resolver:DataConnector>*element) for connecting to the attribute sources,
2. And attribute definitions (*<resolver:AttributeDefinition>* element) that define the attribute type (*xsi:type*) and how it maps to the source (*sourceAttributeID* attribute).

Attribute definitions are associated with a data connector via the *ref* parameter of the *resolver:Dependency* child node.

**Note** For information, see the online help topic [AttributeResolverConfiguration](https://wiki.shibboleth.net/confluence/display/IDP30/AttributeResolverConfiguration)[[10]](#footnote-10) on the Shibboleth Community wiki site.

The Shibboleth IdP software is preconfigured to include a number of assertion attributes in the SAML 2.0 assertions it generates, including an example of *eduPersonScopedAffiliation* and *eduPersonTargetedID*. Here, we will modify the default configuration in the *attribute-resolver.xml*file to add the two above *ImmutableID* and *UserID* attributes as well the data connector for our LDAP AD LDS directory instance *ShibbolethDir*.

To inform Shibboleth of these requirements and configure the above claims type, proceed with the following steps:

1. Use Windows Explorer to navigate to %IDP\_Home%\conf, e.g. C:\Program Files (x86)\Shibboleth\IdP\conf.
2. Right-click the ***attribute-resolver.xml***file, and then click **Edit**. The file should open in Notepad.
3. Scroll until you see the following section:

<!-- ========================================== -->

<!-- Attribute Definitions -->

<!-- ========================================== -->

1. Move one line down and insert the following text:

<!-- Use AD LDS objectGUID for ImmutableID -->

<AttributeDefinition id="ImmutableID" xsi:type="Simple" sourceAttributeID="userPrincipalName">

<Dependency ref="myLDAP" />

</AttributeDefinition>

<!-- mail for Azure AD User ID -->

<AttributeDefinition id="UserId" xsi:type="Simple" sourceAttributeID="mail">

<Dependency ref="myLDAP" />

<AttributeEncoder xsi:type="SAML2String" name="IDPEmail" friendlyName="UserId" />

</AttributeDefinition>

1. Move to the bottom of the file to the line before "</AttributeResolver>”, and insert the following text (copied from attribute-resolve-ldap.xml and the “<LDAPProperty>” element added):

<DataConnector id="myLDAP" xsi:type="LDAPDirectory" ldapURL="%{idp.attribute.resolver.LDAP.ldapURL}" baseDN="%{idp.attribute.resolver.LDAP.baseDN}" principal="%{idp.attribute.resolver.LDAP.bindDN}" principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}" useStartTLS="%{idp.attribute.resolver.LDAP.useStartTLS:true}" connectTimeout="%{idp.attribute.resolver.LDAP.connectTimeout}" trustFile="%{idp.attribute.resolver.LDAP.trustCertificates}" responseTimeout="%{idp.attribute.resolver.LDAP.responseTimeout}"> <FilterTemplate> <![CDATA[ %{idp.attribute.resolver.LDAP.searchFilter} ]]> </FilterTemplate>

<LDAPProperty name="java.naming.ldap.attributes.binary" value="objectGUID" /> <ConnectionPool minPoolSize="%{idp.pool.LDAP.minSize:3}" maxPoolSize="%{idp.pool.LDAP.maxSize:10}" blockWaitTime="%{idp.pool.LDAP.blockWaitTime:PT3S}" validatePeriodically="%{idp.pool.LDAP.validatePeriodically:true}" validateTimerPeriod="%{idp.pool.LDAP.validatePeriod:PT5M}" expirationTime="%{idp.pool.LDAP.idleTime:PT10M}" failFastInitialize="%{idp.pool.LDAP.failFastInitialize:false}" /> </DataConnector>

1. Save and close the ***attribute-resolver.xml*** file.

### Configuring the Shibboleth attribute filter

Shibboleth IdP must be configured to release the two previous required attributes to Azure AD.

The *%IDP\_HOME%\conf\****attribute-filter.xml*** file is used to determine which attributes to release to specific service providers.

The file contains a set of attribute filter policy (*<AttributeFilterPolicy>* element) nodes that define rules (*<PolicyRequirementRule>* element) for allowing a service provider like Azure AD access to the attributes, and attribute filters that define which attributes are released.

It contains a rule which releases the transient ID to all SPs; this rule should be kept in place when you edit the *attribute-filter.xml* file to add your own rules.

**Note** For information, see the online help topic [AttributeFilterConfiguration](https://wiki.shibboleth.net/confluence/display/IDP30/AttributeFilterConfiguration)[[11]](#footnote-11) on the Shibboleth Community wiki site.

To release the attributes to Azure AD, proceed with the following steps:

1. Use Windows Explorer to navigate to %IDP\_Home%\conf, e.g. C:\Program Files (x86)\Shibboleth\IdP\conf.
2. Right-click the ***attribute-filter.xml***file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “urn:mace:shibboleth:2.0:afp http://shibboleth.net/schema/idp/shibboleth-afp.xsd”.
4. Move one line down and insert the following text to modify the list of attributes that will be released:

<AttributeFilterPolicy id="PolicyForWindowsAzureAD">

<PolicyRequirementRule xsi:type="Requester" value="urn:federation:MicrosoftOnline" />

<!-- Release mail as Azure AD User ID -->

<AttributeRule attributeID="UserId">

<PermitValueRule xsi:type="ANY" />

</AttributeRule>

<!-- Release Immutable ID to Azure AD -->

<AttributeRule attributeID="ImmutableID">

<PermitValueRule xsi:type="ANY" />

</AttributeRule>

</AttributeFilterPolicy>

The settings showed above release the *UserId* and *ImmutableID* required attributes only to Azure AD. The settings use specific AttributeFilterPolicy IDs to indicate the attributes are required by Azure AD.

1. Save and close the ***attribute-filter.xml*** file.

### Configuring the Shibboleth persistent id generator

Azure AD uses the persistent nameid and Shibboleth IdP needs to be configured to release the immutableID IdP attribute.

The *%IDP\_HOME%\conf\****saml-nameid.xml*** file is used to configure custom name id formats.

The file contains a set of name Id formatters. By default only the transientId generator is configured. Mappings need to be added to configure a custom persistent id generator.

**Note** For information, see the online help topic [NameIDGenerationConfiguration](https://wiki.shibboleth.net/confluence/display/IDP30/NameIDGenerationConfiguration)[[12]](#footnote-12) on the Shibboleth Community wiki site.

To release the attributes to Azure AD, proceed with the following steps:

1. Use Windows Explorer to navigate to %IDP\_Home%\conf, e.g. C:\Program Files (x86)\Shibboleth\IdP\conf.
2. Right-click the ***saml-namid.xml***file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “shibboleth.SAML2PersistentGenerator”.
4. Move one line down and insert the following text to modify the list of generators that will be released:

<!--

<ref bean="shibboleth.SAML2PersistentGenerator" />

-->

<!-- Persistent ID Generator for all entities except Azure AD -->

<!-- Additional configuration is required if this generator is being used

     <bean parent="shibboleth.SAML2PersistentGenerator">

         <property name="activationCondition">

             <bean parent="shibboleth.Conditions.NOT">

                 <constructor-arg>

                     <bean parent="shibboleth.Conditions.RelyingPartyId" c:candidates="urn:federation:MicrosoftOnline" />

                 </constructor-arg>

             </bean>

         </property>

     </bean>

     <!-- Custom persistent ID Generator for Azure AD -->

     <bean parent="shibboleth.SAML2AttributeSourcedGenerator"

              p:format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"

              p:attributeSourceIds="#{ {'ImmutableID'} }">

         <property name="activationCondition">

             <bean parent="shibboleth.Conditions.RelyingPartyId" c:candidates="urn:federation:MicrosoftOnline" />

         </property>

     </bean>

The settings showed above release the *UserId* and *ImmutableID* required attributes only to Azure AD. The settings use specific AttributeFilterPolicy IDs to indicate the attributes are required by Azure AD.

1. Save and close the ***saml-nameid.xml*** file.

### Restarting Shibboleth IdP and verifying functionality

To restart Shibboleth IdP and verify functionality, proceed with the following steps:

1. Follow the steps as per section § Restarting the Shibboleth web server to take into account the updated configuration.
2. Follow the steps as per section § Ensuring that the Shibboleth IdP is running.
3. Follow the steps as per section § Performing configuration tests.

These steps stop and start Apache Tomcat to restart Shibboleth IdP and ensure the updated XML files are loaded. Shibboleth may fail to start if there is a problem with one or more of the configuration files.

If you encounter any issue, check Jetty and Shibboleth’s log files after restart, located at:

* %JETTY\_HOME%\logs\
  + DATE.stderrout.log
* %IDP\_HOME%\logs\
  + idp-process.log

If you are still stumped, please check out [Shibboleth troubleshooting](https://wiki.shibboleth.net/confluence/display/IDP30/Troubleshooting)[[13]](#footnote-13) on the Shibboleth Community wiki.

## Installing Azure Active Directory Module for PowerShell (MSOnline)

A PowerShell module is a package that contains commands, cmdlets, providers, functions, variables, aliases, and more.

**Note** PowerShell is a task-based command-line shell and scripting language designed for system administration. Unlike most shells, which accept and return text, PowerShell is built on top of the .NET Framework, and accepts and returns .NET Framework objects. PowerShell introduces the concept of a cmdlet (pronounced "command-let"), a simple, single-function command-line tool built into the shell. Cmdlets have the following naming convention: a verb and noun separated by a dash (-), such as Get-Help, Get-Process, and Start-Service. PowerShell includes more than one hundred basic core cmdlets. For more information about PowerShell, see [Getting Started with PowerShell](https://docs.microsoft.com/en-us/powershell/scripting/getting-started/getting-started-with-windows-powershell?view=powershell-6).

The Azure Active Directory Module for PowerShell (MSOnline) cmdlets can be used for Azure AD administrative tasks such as user management, domain management, and configuration of single sign-on. This topic includes information about how to install these cmdlets, and use them set up single sign-on to Azure AD and Azure AD protected resources.

**Note** For additional information, see [Azure Active Directory Module for PowerShell (MSOnline)](https://docs.microsoft.com/en-us/powershell/azure/active-directory/overview?view=azureadps-1.0)[[14]](#footnote-14) and [MSOnline cmdlet reference](https://docs.microsoft.com/en-us/powershell/module/msonline/?view=azureadps-1.0).

The Azure AD Module is supported on the following Windows operating systems with the Microsoft .NET Framework 4.5 and above and PowerShell 3.0 and newer: Windows 10, Windows 8.1 Pro, Windows 8.1 Enterprise, Windows 7 SP1, Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012 R2 SP1.

**Note** Administrative privileges are needed on the local computer to install the Azure Active Directory Module for PowerShell (MSOnline)[[15]](#footnote-15).

To install the Azure Active Directory Module for PowerShell (MSOnline), do the following:

1. Launch PowerShell using the Run as Administrator option,
2. At the PowerShell prompt type **Install-Module -Name MSOnline** and press **Enter**.

PowerShell will install the MSOnline cmdlets for Azure Active Directory from the PowerShell Gallery.

## Connecting PowerShell to Azure AD

Before you can run any of the cmdlets discussed in this article, you must first connect to your online service. To do so, run the cmdlet **Connect-MsolService** at the PowerShell command prompt. You will then be prompted for your credentials. If you want, you can supply your credentials in advance, for example:

PS C:\Windows\system32>$Msolcred = Get-credential

PS C:\Windows\system32>Connect-MsolService -Credential $MsolCred

The first command prompts for credentials and stores them as $Msolcred. The next command uses those credentials as $Msolcred to connect to the service. When prompted, enter the Global Administrator account credentials of your Azure AD directory, e.g. [admin@contoso.onmicrosoft.com](mailto:admin@contoso.onmicrosoft.com).



To connect to a specific environment of Azure Active Directory, use the AzureEnvironment parameter, as follows:

PS C:\Windows\system32> Connect-MsolService -AzureEnvironment "AzureGermanyCloud"

This example connects your PowerShell session to the German Azure AD environment.

## Setting up a trust between Shibboleth IdP and Azure AD

Before configuring single sign-on, Azure AD must have at least one custom domain available. You cannot configure the default domain that is provided by Microsoft as a single sign-on domain. The default domain from Microsoft ends with “onmicrosoft.com”.

Each on-premises LDAP domain or subdomain that you want to use for single sign-on must be available in Azure AD as a custom domain.

Azure AD custom domains are “federated” using the Azure Active Directory Module for PowerShell (MSOnline). You can use this module to run a series of cmdlets in PowerShell to add or convert domains for single sign-on.

Additionally, custom domains can be added as “Standard” via the Azure Portal or Office 365 Admin Center GUI and later converted to “Federated” using PowerShell.

Each Azure AD custom domain that you want to configure for single sign-on must either be *added* as a “Federated” domain or *converted* to be a single sign-on domain from a “Standard” domain in Azure AD. Adding or converting a domain sets up a trust between your Shibboleth IdP and Azure AD.

**Note** Single sign-on is configured per each Azure AD custom domain or subdomain. It’s possible, for example, to have one FQDN redirect to Shibboleth Identity Provider, one FQDN redirect to Active Directory Federation Services, and multiple, separate Azure AD custom domains or subdomains redirect to one Shibboleth IdP.

### Adding a new custom domain to Azure AD (REVISION IS UNDERWAY)

**Note** For now, recommend using the Office 365 Admin center or Azure Portal web GUI to add new custom domains unless there is a requirement to add tens or hundreds of domains

After signing up for Azure Active Directory or Office 365, the only domain associated with your subscription account is the default tenant domain chosen during registration, e.g. *idmgt14.onmicrosoft.com*, and this default domain in the onmicrosoft.com namespace cannot be used for single sign-on.

Consequently, we start by creating a standard domain for the LDAP *shib.idmgt.contoso.com* domain.

To create a standard (managed) domain, proceed with the following steps:

1. Connect PowerShell to Azure AD (see § Connecting Windows PowerShell to Azure AD ).

This cmdlet connects you to the Cloud service. Creating a context that connects you to the cloud service is required before running any of the additional cmdlets installed by the tool.

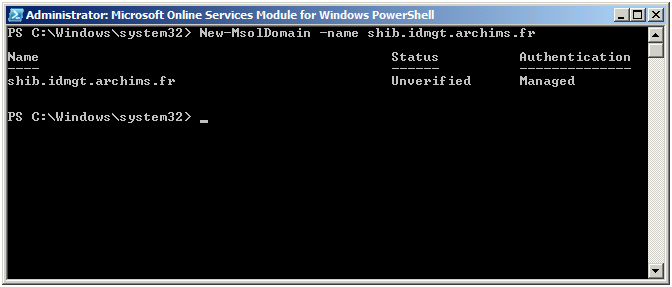
1. Create a new standard domain for the *shib.idmgt.contoso.com* domain with the following command:

PS C:\Windows\system32> New-MsolDomain –name shib.idmgt.contoso.com

Name Status Authentication

---- ------ --------------

shib.idmgt.contoso.com unverified Managed



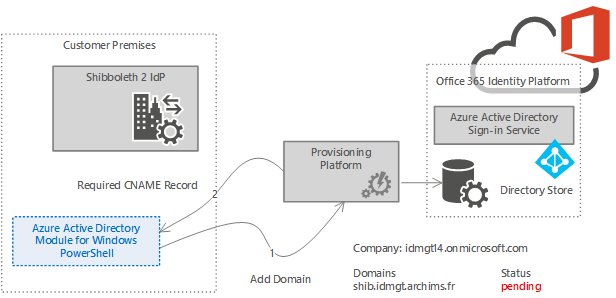


Figure 3 Standard domain creation

1. Get the DNS record information to create for the new managed domain with the following command:

PS C:\Windows\system32> Get-MsolDomainVerificationDns –DomainName shib.idmgt.contoso.com

canonicalName : ps.microsoftonline.com

ExtensionData : System.Runtime.Serialization.ExtensionDataObject

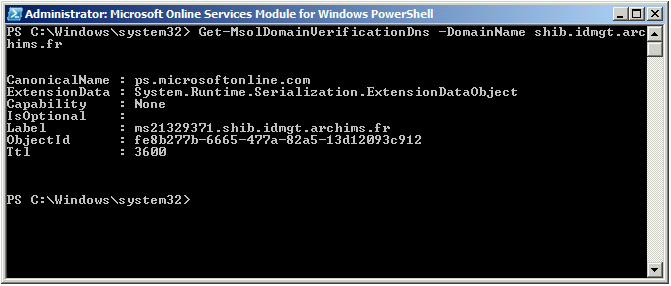
Capability : None

IsOptional :

Label : ms21329371.shib.idmgt.contoso.com

ObjectId : fe8b277b-6665-477a-82a5-13d12093c912

Ttl : 3600



1. To prove that you control the domain, use the output of the above command to create a CNAME record in the DNS server of the domain used previously. The name of the record should match the Label value and the value of the record should match the CanonicalName output above:

*Name: ms21329371.shib.idmgt.contoso.com*

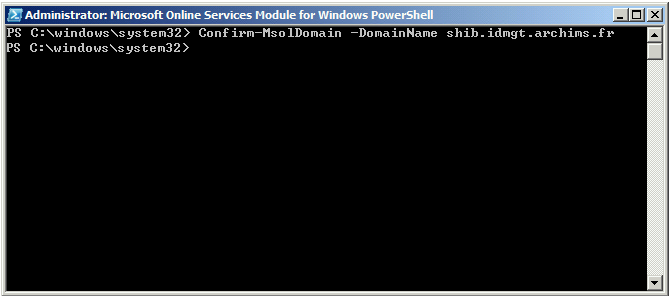
*Type: CNAME*

*Value: ps.microsoftonline.com*

Azure AD indeed uses a DNS record that you create at your registrar to confirm that you own the domain. For additional information, please refer to the Microsoft TechNet articles [Add your domain](http://technet.microsoft.com/en-us/library/hh969247.aspx)[[16]](#footnote-16) and [Verify a domain at any domain name registrar](http://technet.microsoft.com/en-us/library/jj151803.aspx)[[17]](#footnote-17).

1. Prove your control of the domain by running the following command:

PS C:\Windows\system32> Confirm-MsolDomain –DomainName shib.idmgt.contoso.com



This verifies the domain proof of ownership.

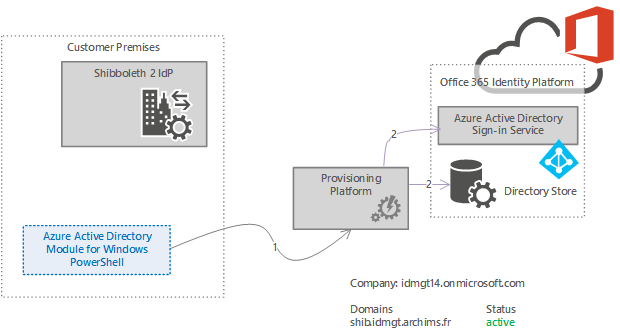


Figure 4 Standard domain verification

### Converting an existing domain for use with single sign-on

As previously mentioned, each on-premises LDAP domain that you want to enable for single sign-on using Shibboleth must either be added as a single sign-on domain or converted to be a single sign-on domain from a standard domain. Adding or converting a domain sets up a trust between Shibboleth IdP and Azure AD.

This is done using the Azure Active Directory Module for PowerShell (MSOnlin) **Set-MsolDomainAuthentication** cmdlet.

This cmdlet has the following arguments:

|  |  |
| --- | --- |
| **Argument** | **Description** |
| -DomainName <string> | The fully qualified domain name (FQDN) to update. |
| -FederationBrandName <string> | The name of the string value shown to users when signing in to Microsoft Online Services. We recommend that customers use something that is familiar to them, such as "Contoso, Inc." |
| -Authentication <DomainAuthenticationType> | The authentication type (managed/federated) of the domain. All users created on this domain will have this authentication type. |
| -PassiveLogOnUri <string> | The URL that web-based clients will be directed to when signing in to Microsoft Online Services. |
| -SigningCertificate <string> | The current certificate used to sign tokens passed to the Microsoft Online Identity platform. |
| -IssuerUri <string> | The unique identifier of the domain in the Azure AD identity platform derived from the federation server. |
| -ActiveLogOnUri <string> | A URL that specifies the end point used by active clients when authenticating with domains set up for single sign-on (also known as identity federation) in Microsoft Online. |
| -LogOffUri <string> | The URL clients are redirected to when they sign out of Microsoft Online Services. |
| -PreferredAuthenticationProtocol <string> | The abbreviation of the federation protocol used to interact with the Microsoft Online Identity platform: SAMLP or WSFED. |

The value of the *IssuerUri* parameter MUST match the *provider* value defined in the *idp.entityID* property in the *%IDP\_HOME%\conf\idp.properties* file.

To allow users to have SSO with Azure AD or its protected resources, e.g. Office 365 or Microsoft Azure, this supposes to convert the on-premises LDAP domain declared in the previous section, i.e. *shib.idmgt.contoso.com* in our configuration, as a “Federated” domain.

The following procedure walks you through converting an existing standard domain to a single sign-on domain.

**Note** Domain may experience an outage that impacts users up to 2 hours after you take this step.

To convert the domain as a federated domain, proceed with the following steps:

1. Connect PowerShell to Azure AD (see § Connecting Windows PowerShell to Azure AD.

This cmdlet connects you to the cloud service. Creating a context that connects you to the cloud service is required before running any of the additional cmdlets installed by the tool.

1. Run the following commands to convert an existing domain (in this example, *shib.idmgt.contoso.com*) for single sign on:

PS C:\Windows\system32> $dom = "shib.idmgt.contoso.com”

PS C:\Windows\system32> $fedBrandName = “IDMGT Shibboleth”

PS C:\Windows\system32> $url = "https://idp0.shib.idmgt.contoso.com/idp/profile/SAML2/POST/SSO"

PS C:\Windows\system32> $uri = “https://idp0.shib.idmgt.contoso.com/idp/shibboleth”

PS C:\Windows\system32> $logoutUrl = “https://idp0.shib.idmgt.contoso.com/idp/profile/Logout”

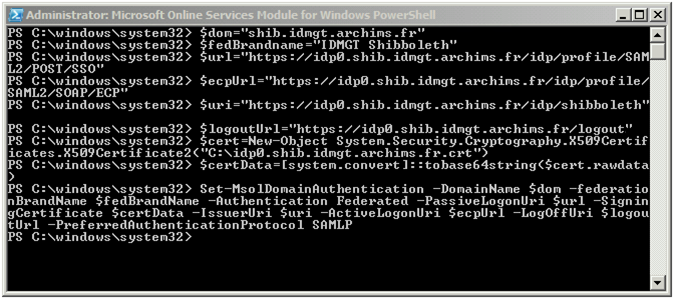
PS C:\Windows\system32> $cert = New-Object

System.Security.Cryptography.X509Certificates.X509Certificate2(

"C:\Program Files (x86)\Shibboleth\IdP\credentials\idp-signing.crt")

PS C:\Windows\system32> $certData = [system.convert]::tobase64string($cert.rawdata)

PS C:\Windows\system32> Set-MsolDomainAuthentication –DomainName $dom –federationBrandName $FedBrandName -Authentication Federated -PassiveLogOnUri $url -SigningCertificate $certData -IssuerUri $uri -LogOffUri $logoutUrl -PreferredAuthenticationProtocol SAMLP



**Note** The $certData is directly source from the credentials/idp-signing.crt.

After the above steps are completed, you can verify that the domain was added correctly and is federated via the Azure Portal or Office 365 Admin center. When you are in the portal, just select the **Admin** option at the top in the navigation bar. Then, in the left column, select the **Domains** under **User** **Management**, then select the domain that you just added and you will see that it is “Federated.”

When the domain is “Federated”, you will no longer have the option to add the domain suffix to the Azure AD user accounts. The users will need to be created on-premises for them to have the federated domain name available to them. You can still create accounts directly in the cloud, but they cannot have the federated domain name assigned to them unless they are created on-premises.

## Provision user principals to Azure AD

Before you can authenticate your users to Azure AD protected resources, you must provision Azure AD with user principals that correspond to the assertion in the SAML 2.0 claim. If these user principals are not known to Azure AD in advance, then they cannot be used for single sign-on with federated custom domains. Various methods can be used to provision user principals, such as Azure AD Connect, Microsoft Identity Manager (MIM) 2016, PowerShell and Microsoft Graph can be used to provision user principals.

Azure AD Connect can be used to provision user principals to your Azure AD directory from Windows Server Active Directory. MIM 2016, Azure Active Directory Module for PowerShell (MSOnline) and Microsoft Graph might be used to automate adding new users to Azure AD and to synchronize changes from non-AD sources, e.g. LDAP directories.

Considering the number of situations to consider in Windows Server Active Directory environments (mono forest, “simple” multi-forest, “complex”, multi-forest) as well as non-AD environments, these advanced topics are out of scope for this document.

For purpose of testing Azure AD single sign-on with your Shibboleth IdP, you can create new Azure AD federated using the Azure Active Directory Module for PowerShell (MSOnline).

This procedure shows how to add a single user to Azure AD.

1. Connect to your Office 365 tenant as a tenant administrator: Connect-MsolService.
2. Create a new user principal:

PS C:\Windows\system32> New-MsolUser `

-UserPrincipalName elwoodf1@contoso.com

-ImmutableId ABCDEFG1234567890

-DisplayName "Elwood Folk"

-FirstName Elwood

-LastName Folk

-AlternateEmailAddresses "Elwood.Folk@contoso.com"

-LicenseAssignment "idmgt14:ENTERPRISEPACK"

-UsageLocation "US"

You can now use this user to verify the single sign-on with Shibboleth IdP.

This simple approach is not suitable for a production environment. Implementing a process to provision user principals and manage the long-term consistency and parity of state between source objects and their representation in Azure AD is a critically important decision to make prior going into production.

**Note** The “UserPrinciplName” value must match the value that you will send for “IDPEmail” in your SAML 2.0 claim and the “ImmutableID” value must match the value sent in your “NameID” assertion.

## Verifying single sign-on with Shibboleth IdP

It is always better, when verifying (and/or) troubleshooting the single sign-on (SSO), to keep it as simple as possible.

Even if an encountered issue concerns, for instance, Exchange Online access, it is better just accessing the Office 365 portal (or the Azure Portal) with the on-premises credentials to verify if the SSO is working. This will allow you to verify if the issue is application/service specific or if the issue is with SSO. If the user can log in to the Office 365 portal but cannot log into OWA with the corporate credentials then you can be sure the issue is not related to SSO.

**Note** For converted domains, it may (but does not usually) take up to 24 hours to set up single sign-on.

To verify the SSO with the Office 365 portal, proceed as follows:

1. Open a browsing session, and then navigate to <https://portal.office.com> to access the Office 365 portal. You will see you are immediately redirected to the *login.microsoftonline.com*.
2. Sign in using the same logon name that you use for your corporate credentials.

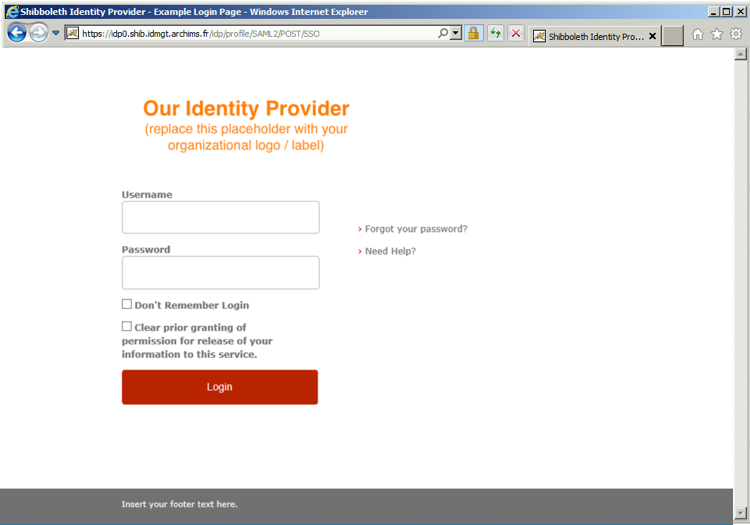
Username: user1@shib.idmgt.contoso.com

1. Click inside **Password**. This triggers a home realm discovery (HRD) process for federated identities to see if the domain part of the UPN is actually federated.

**Note** If you turn on HTTP tracing on Internet Explorer or observe the traffic via a tool like the Telerik [Fiddler](http://www.telerik.com/fiddler)[[18]](#footnote-18) HTTP trace application, you can see that the login.microsoftonline.com URL is calling GetUserRealm as part of the home realm discovery (HRD) process. You will also notice that the results show the Shibboleth IdP passive endpoint information.

Likewise, with Firefox, you can use the [SAML tracer](https://addons.mozilla.org/en-US/firefox/addon/saml-tracer/)[[19]](#footnote-19), a Firefox plugin that allows you to trace and review all front-channel SAML 2.0 messages sent as you browse web pages.

1. If SSO is correctly set up, you will notice that the user cannot even type their password. You’ll be redirected to the Shibboleth IdP passive endpoint with the defined Username/Password login page.



1. Enter the on-premises corporate credentials for the test user user1:

*Username: User1*

*Password: Password1*

**Note** No action is needed by the Admin to enable existing users to access their email. The ImmutableID of the user will be passed to Azure AD in the SAML 2.0 token.

**Note** To customize the default Velocity Template login page, see the online help topic [AuthenticationConfiguration](https://wiki.shibboleth.net/confluence/display/IDP30/AuthenticationConfiguration)[[20]](#footnote-20) on the Shibboleth Community wiki.

If you are able to sign in, the single sign-on has been set up correctly.

## Troubleshooting the single sign-on (SSO) with Shibboleth IdP

If you run into issues on the Shibboleth IdP side, you may wish to check Tomcat and Shibboleth’s log files, located under *%JETTY\_BASE%\logs\* and *%IDP\_HOME%\logs\.*

If you are still stumped, please check out [troubleshooting](https://wiki.shibboleth.net/confluence/display/IDP30/Troubleshooting)[[21]](#footnote-21) at the Shibboleth Community wiki site.

Additionally, Microsoft has provided a tool that you can use to test your SAML 2.0 based identity provider available at <https://testconnectivity.microsoft.com/?tabid=Client>. Before running the test tool you must have configured an Azure AD tenant to federate with your identity provider.

# Understanding how federated authentication works in Office 365

This section aims at providing additional information on the configuration (via the modification of the Shibboleth IdP configuration files and the use of the Azure Active Directory Module for PowerShell) in order to setup single sign-on. It focuses on an explanation of the resulting settings on the IdP as well as the several types of interaction between the key components involved in the transaction, i.e. the client, the on-premises Shibboleth infrastructure, Azure AD and its protected resources, such as Office 365 and Microsoft Azure.

## Understanding the Shibboleth IdP configuration (REVISION OF THIS SECTION REQUESTED)

As covered in section $ 4.3.1 Adding Azure AD as a relying party, adding a partner like Azure AD into a Shibboleth IdP is mainly done by referencing the partner’s XML metadata.

Azure AD publishes its federation metadata at the following URL:

<https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml>.

You should always check for the latest Azure AD metadata. Here is the current value of the metadata:

<?xml version="1.0" encoding="utf-8"?>

<EntityDescriptor ID="\_0c0d1ca7-7292-4bc6-801c-f880f6098f4e"

entityID="urn:federation:MicrosoftOnline"

xmlns="urn:oasis:names:tc:SAML:2.0:metadata"

xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport">

<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">

<SignedInfo>

<CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>

<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>

<Reference URI="#\_0c0d1ca7-7292-4bc6-801c-f880f6098f4e">

<Transforms>

<Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>

<Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>

</Transforms>

<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<DigestValue>lmuywskSIZK9HjyNuvYE+Y2vtNU=</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>WbCHKG4bcAcRzKQIdNIuVQqdHgfIpwpgH8RP5frAvf7+SZIxN3JIMW6FyX+YyQoPO8RV9YcBL8uTWFFD2xJ1TafogDFyFE8gnzhZAw

Sppj4DO60v/vTxnjR3gEucdaJIfqfDIAaORmgSLurHdlGuhq/LZCjZ3d6lsnDV1pHbSjt/vDM1okfBk0O9RqIsh8UozdXaqrY8isNK

jD3lUtANYPYTiJGiK9LeJd+UkzTjbluWxRMt4qaxWp8d92DeD/do/99cKY8Xy5uFsM2qXDHNqM6IH17HWJWMP7oQN53NtnVDq+8Pqa

bF2q6Ai8s/NMJUvoYN/ELCIUZ57pD+1A74WQ==

</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>MIIDYDCCAkigAwIBAgIJALLJPAyvf2sjMA0GCSqGSIb3DQEBBQUAMCkxJzAlBgNV

BAMTHkxpdmUgSUQgU1RTIFNpZ25pbmcgUHVibGljIEtleTAeFw0xNDA3MTgxOTUz

NDBaFw0xOTA3MTcxOTUzNDBaMCkxJzAlBgNVBAMTHkxpdmUgSUQgU1RTIFNpZ25p

bmcgUHVibGljIEtleTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANYD

KgByFZdqtTnnpF4IfIp4i2XLg2rLIo+mu4DmW9gRLlBJCNc7YESUxpKzuFYaANd8

fWsDigJZTXbhOQApSpw4xXFnor2vJ1zm94LtqjcVEXTjUml5gAIS4pwuOU3ZfO/0

eTG0gDYp4a0L/mzzTRsnwe/8WMPIE75Bq2zAyAZ9aePvl3QX7cXYLPfeK4QTgK3B

5lwe1wWu3y5oQidjcSok8Frf80xzuCYuOa+ZUK3JibpLLCrT4uwiqf+KREDSdc4b

PPlq0PWI4sQr1tha8yypRSvOH+/MxcfSRSnl6Uc+gm8nVEEWWIu4hhu6NIfG91mM

UqJuzkgLCi6Gov6JS8UCAwEAAaOBijCBhzAdBgNVHQ4EFgQUnQoq7sI3R8rde4sQ

s6nGEbJm3LcwWQYDVR0jBFIwUIAUnQoq7sI3R8rde4sQs6nGEbJm3LehLaQrMCkx

JzAlBgNVBAMTHkxpdmUgSUQgU1RTIFNpZ25pbmcgUHVibGljIEtleYIJALLJPAyv

f2sjMAsGA1UdDwQEAwIBxjANBgkqhkiG9w0BAQUFAAOCAQEAf4jaNhKzRG3k+52W

oM9nnISP7rlWIeWwH6EQGUlF6ozSP/03gYMAdqpdhww5zNwKzi7TQVbDC0pgq/tq

zHv6JEI0R4B6h7/TJ1pYPxdvIFQrE27RHESltH/m+5UkVnayLqRD3/fi4zf4aEpx

SDZ73MCR5LanPGqvlAMz29AL3g1ynj+eu7xMfFsM/8+qJaCXuxT5/30eeLEe+PYi

kA/PhEwp+qkDQWPvdAwEghuUaFvtKAgDZierjpGzHZnYkXTTDTHVe1iP7tsAJH5q

K3qdcv3UGPyZrjC/lietJcAcnwVoZQ93v2ieGfcKKN+PFN9M59/BkPo62HPoGNNx

2ZDQaQ==

</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

<Extensions>

<alg:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<alg:SigningMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>

</Extensions>

<SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol" WantAssertionsSigned="true">

<KeyDescriptor use="signing">

<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

<ds:X509Data>

<ds:X509Certificate>MIIDYDCCAkigAwIBAgIJALLJPAyvf2sjMA0GCSqGSIb3DQEBBQUAMCkxJzAlBgNV

BAMTHkxpdmUgSUQgU1RTIFNpZ25pbmcgUHVibGljIEtleTAeFw0xNDA3MTgxOTUz

NDBaFw0xOTA3MTcxOTUzNDBaMCkxJzAlBgNVBAMTHkxpdmUgSUQgU1RTIFNpZ25p

bmcgUHVibGljIEtleTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANYD

KgByFZdqtTnnpF4IfIp4i2XLg2rLIo+mu4DmW9gRLlBJCNc7YESUxpKzuFYaANd8

fWsDigJZTXbhOQApSpw4xXFnor2vJ1zm94LtqjcVEXTjUml5gAIS4pwuOU3ZfO/0

eTG0gDYp4a0L/mzzTRsnwe/8WMPIE75Bq2zAyAZ9aePvl3QX7cXYLPfeK4QTgK3B

5lwe1wWu3y5oQidjcSok8Frf80xzuCYuOa+ZUK3JibpLLCrT4uwiqf+KREDSdc4b

PPlq0PWI4sQr1tha8yypRSvOH+/MxcfSRSnl6Uc+gm8nVEEWWIu4hhu6NIfG91mM

UqJuzkgLCi6Gov6JS8UCAwEAAaOBijCBhzAdBgNVHQ4EFgQUnQoq7sI3R8rde4sQ

s6nGEbJm3LcwWQYDVR0jBFIwUIAUnQoq7sI3R8rde4sQs6nGEbJm3LehLaQrMCkx

JzAlBgNVBAMTHkxpdmUgSUQgU1RTIFNpZ25pbmcgUHVibGljIEtleYIJALLJPAyv

f2sjMAsGA1UdDwQEAwIBxjANBgkqhkiG9w0BAQUFAAOCAQEAf4jaNhKzRG3k+52W

oM9nnISP7rlWIeWwH6EQGUlF6ozSP/03gYMAdqpdhww5zNwKzi7TQVbDC0pgq/tq

zHv6JEI0R4B6h7/TJ1pYPxdvIFQrE27RHESltH/m+5UkVnayLqRD3/fi4zf4aEpx

SDZ73MCR5LanPGqvlAMz29AL3g1ynj+eu7xMfFsM/8+qJaCXuxT5/30eeLEe+PYi

kA/PhEwp+qkDQWPvdAwEghuUaFvtKAgDZierjpGzHZnYkXTTDTHVe1iP7tsAJH5q

K3qdcv3UGPyZrjC/lietJcAcnwVoZQ93v2ieGfcKKN+PFN9M59/BkPo62HPoGNNx

2ZDQaQ==

</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</KeyDescriptor>

<KeyDescriptor use="signing">

<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

<ds:X509Data>

<ds:X509Certificate>MIIDYDCCAkigAwIBAgIJAKLDsqkylLefMA0GCSqGSIb3DQEBBQUAMCkxJzAlBgNV

BAMTHkxpdmUgSUQgU1RTIFNpZ25pbmcgUHVibGljIEtleTAeFw0xNDEwMTAxODE2

MTNaFw0xOTEwMDkxODE2MTNaMCkxJzAlBgNVBAMTHkxpdmUgSUQgU1RTIFNpZ25p

bmcgUHVibGljIEtleTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAM7A

3m6uvOxEsX+NlB1hnflaR8DJj597wY3qyh/FX4O6rKvU2leAfINmBWcjEFApCKi9

p5uIaZpNlDpPQ+R3BaZx+4NhHbOMpeWlpIiZHL61lwbulzurffUPhtzQNHAVzOBk

ZsOgN9BD/hOleU//d+IXz08ateUb3Ip2vyaodilYQDDi5M9yOhanv1cO1Usjo2xT

LfiK+TVygu+8bo+/8JHGPRy6pnghng970DRBDkVrKzozlrnmMesdSrtuCnsgyRbE

XckxaQ8S2nDYyFqBI0PkcBW8+0akdFWW58Os5cGbPFeHi6vtZCR5pWw5pnqtuoip

rdk9jg1axT3vwu+RVdcCAwEAAaOBijCBhzAdBgNVHQ4EFgQUBjNylGJBvkAY/4yI

IoD00R6p5hIwWQYDVR0jBFIwUIAUBjNylGJBvkAY/4yIIoD00R6p5hKhLaQrMCkx

JzAlBgNVBAMTHkxpdmUgSUQgU1RTIFNpZ25pbmcgUHVibGljIEtleYIJAKLDsqky

lLefMAsGA1UdDwQEAwIBxjANBgkqhkiG9w0BAQUFAAOCAQEAQGZUlJ3zzJvy1OLd

tV3NTYHlbVHm3Fty17xqW9Ui8GE8sEWeUdHA6eURNNpNpd+gAGC6Tp+k+cU1LlPw

Xm7BAATJ/2DjY8tzRc6r6EneQWRkIa8xpbvknXvUml6iFgo2ofOWLaFk6XpQ64MA

O35wv9XEARNabJ9wJSRSevUigAx2U2GvaorV5PgqHImiKTSrL0K6j8B4OqXWUqP0

KGf7pCdGlrq2XEl95N2zj8n/scvA9JasImztsVlZ+WxeF+OAMvWQQFc54gC6lwWc

8kno8vPn3lwxVkTU0o9wcHnOhNi2hzVDV85sz7P9dOZYF73uy1uLshdjCcwlmQ2l

A9OV9w==

</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</KeyDescriptor>

<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"

Location="https://login.microsoftonline.com/login.srf"/>

<NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</NameIDFormat>

<NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>

<NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</NameIDFormat>

<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>

<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</NameIDFormat>

<AssertionConsumerService isDefault="true"

index="0"

Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"

Location="https://login.microsoftonline.com/login.srf"/>

<AssertionConsumerService index="1"

Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"

Location="https://login.microsoftonline.com/login.srf"/>

<!-- PAOS functionality is NOT supported by this service. The binding is only included to ease setup and integration

with Shibboleth ECP -->

<AssertionConsumerService index="2"

Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS"

Location="https://login.microsoftonline.com/login.srf"/>

</SPSSODescriptor>

</EntityDescriptor>

The general syntax and semantics of metadata are defined in the [Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0](http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf)[[22]](#footnote-22) (SAMLMeta) document. It covers the configuration data (endpoint URLs, key material for verifying signatures, etc.) to establish trusts between SAML 2.0 entities.

The *<SPSSODescriptor>* element corresponds to the relying party role of Azure AD in which we are interested.

This element defines two SAML 2.0 based endpoints for Azure AD for the interaction with the organization’s on-premises Shibboleth infrastructure:

1. A passive endpoint for web clients (browser) based on the SAML 2.0 web Browser SSO profile and the HTTP-POST binding: “*urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST*”.

The related authentication flow is described in section § Understanding the passive/web profile authentication flow.

1. An active endpoint for rich e-mail client clients (Outlook 2010, Thunderbird 2.0, etc.) based on the Enhanced Client or Proxy (ECP) profile along with the HTTP-POST-SimpleSign binding: “*urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign*”.

The related authentication flow is described in section § Understanding the ECP profile authentication flow.

These endpoints are located at the same URL, i.e. <https://login.microsoftonline.com/login.srf>.

This information serves to define Azure AD as a relying party in the Shibboleth IdP configuration. The configuration metadata is loaded into the Shibboleth IdP by metadata providers. Metadata providers are configured in the *metadata-provider.xml* file, which defines how the Shibboleth IdP should interact with service providers such as Azure AD in the federation and how it gets the federation metadata. This file may only contain one top-level provider. By default, the top level provider is a chaining provider that contains other metadata providers and uses them in the order defined.

As previously illustrated, we have opted, with the definition of a file-backed HTTP metadata provider to directly locate and load the Azure AD metadata via HTTP and to back up it to a local file.

“*If federation is broken. It's PKI. If it is not PKI, there's a typo. If you typed it correctly (case counts!). It's PKI*”

Laura E. Hunter

The metadata definition helps preventing such issues. The *cacheDuration* parameter of the file-backed HTTP metadata provider enables to define the maximum time between the metadata refresh automatically occurs. This seamlessly takes into account any change that occurs on the Azure AD platform side. Such a capability greatly lessens the administrative effort to maintain the relying party trust on the Shibboleth IdP side.

Conversely, as described in section § Federation metadata defined, the Shibboleth IdP publishes its own metadata:

<?xml version="1.0" encoding="UTF-8"?>

<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"

entityID="https://idp0.shib.idmgt.contoso.com/idp/shibboleth"

xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:shibmd="urn:mace:shibboleth:metadata:1.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<IDPSSODescriptor protocolSupportEnumeration="urn:mace:shibboleth:1.0 urn:oasis:names:tc:SAML:1.1:protocol

urn:oasis:names:tc:SAML:2.0:protocol">

<Extensions>

<shibmd:Scope regexp="false">

shib.idmgt.contoso.com

</shibmd:Scope>

</Extensions>

<KeyDescriptor>

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>

MIIFpzCCBI+gAwIBAgIKa2DcuwABAAAM+jANBgkqhkiG9w0BAQUFADCBgDETMBEG

CgmSJomT8ixkARkWA2NvbTEZMBcGCgmSJomT8ixkARkWCW1pY3Jvc29mdDEUMBIG

CgmSJomT8ixkARkWBGNvcnAxFzAVBgoJkiaJk/IsZAEZFgdyZWRtb25kMR8wHQYD

VQQDExZNU0lUIE1hY2hpbmUgQXV0aCBDQSAyMB4XDTEyMTAxMTE5MzE0M1oXDTE0

MTAxMTE5MzE0M1owJTEjMCEGA1UEAxMaaWRwMC5zaGliLmlkbWd0LmFyY2hpbXMu

ZnIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCQOl+uVBg+9WqjZljy

HHIaKs1ptpMVB/dkElLaVI8/v+FuqZWeQNTH5OsQxi8XUrynsZA+9x8/r/ZFBqSu

2rK8IAZsFq4WW2uxSza+k4FaJMMShyS5rfdlxq/hnwwsmr8anJoD72xLVlygEbnu

rgzIj2wwqYfg0rKw6D1hEYTkVTqJwHNIvMQXitMVA8nlMkRMLAQUuSBmnwgyvB8G

7G4js5qi8JGrGOKHEfTW3CDE5xWUI7+3Rq6s+4F2mvPAgoXDuZITfHY5E8skRdYo

xS71yiblHcg2OIT3Vfe5iXCBOcvgQ8pv+bxEbQRe/C8CVEPIoasRsVNFYSDHudJT

2+x/AgMBAAGjggJ7MIICdzAdBgNVHQ4EFgQU8K9spMJWL6k2wdoi0PXN8MdWwkMw

CwYDVR0PBAQDAgSwMB8GA1UdIwQYMBaAFOvbEV74CZ7Y1mKc/WKd44RKKOEnMIHu

BgNVHR8EgeYwgeMwgeCggd2ggdqGT2h0dHA6Ly9tc2NybC5taWNyb3NvZnQuY29t

L3BraS9tc2NvcnAvY3JsL01TSVQlMjBNYWNoaW5lJTIwQXV0aCUyMENBJTIwMigx

KS5jcmyGTWh0dHA6Ly9jcmwubWljcm9zb2Z0LmNvbS9wa2kvbXNjb3JwL2NybC9N

U0lUJTIwTWFjaGluZSUyMEF1dGglMjBDQSUyMDIoMSkuY3JshjhodHRwOi8vY29y

cHBraS9jcmwvTVNJVCUyME1hY2hpbmUlMjBBdXRoJTIwQ0ElMjAyKDEpLmNybDCB

rQYIKwYBBQUHAQEEgaAwgZ0wVQYIKwYBBQUHMAKGSWh0dHA6Ly93d3cubWljcm9z

b2Z0LmNvbS9wa2kvbXNjb3JwL01TSVQlMjBNYWNoaW5lJTIwQXV0aCUyMENBJTIw

MigxKS5jcnQwRAYIKwYBBQUHMAKGOGh0dHA6Ly9jb3JwcGtpL2FpYS9NU0lUJTIw

TWFjaGluZSUyMEF1dGglMjBDQSUyMDIoMSkuY3J0MD8GCSsGAQQBgjcVBwQyMDAG

KCsGAQQBgjcVCIPPiU2t8gKFoZ8MgvrKfYHh+3SBT4PC7YUIjqnShWMCAWQCAQow

HQYDVR0lBBYwFAYIKwYBBQUHAwIGCCsGAQUFBwMBMCcGCSsGAQQBgjcVCgQaMBgw

CgYIKwYBBQUHAwIwCgYIKwYBBQUHAwEwDQYJKoZIhvcNAQEFBQADggEBAHrGC1bh

ajRwByTdibBK7uDIKphDIdb6ReWkBZ0XckGSRNu447gjfNoTA6KOldvHj+Qv03IV

hVjQNtLmD5NTlE3qmiDJ2Rt23k1JQ5ixZAuJRXw8Xa7Muqx5Lxov2N+CoNw9ZPpz

fXQXOL3TiWE4KiLMlvcNSUMFxmO0P3gwpFxj8MrOb5WATHOxbqItqQejqPZCfbqI

Penb+7uL1HNrbPlpyawZpB8IyLqITlkn4I8ihk75aV6mtWjNinnP2umjXYg+7x8e

lIx/JgddkWNWNTsQm4XtVbCnjunyp68SAX6OFoOBHTI2uhwYx7EB8c02ltOlsXxz

gbKSVzXZ2oam1ac=

</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</KeyDescriptor>

<ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"

Location="https://idp0.shib.idmgt.contoso.com:8443/idp/profile/SAML1/SOAP/ArtifactResolution"

index="1"/>

<ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"

Location="https://idp0.shib.idmgt.contoso.com:8443/idp/profile/SAML2/SOAP/ArtifactResolution"

index="2"/>

<NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>

<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>

<SingleSignOnService Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"

Location="https://idp0.shib.idmgt.contoso.com/idp/profile/Shibboleth/SSO"/>

<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"

Location="https://idp0.shib.idmgt.contoso.com/idp/profile/SAML2/POST/SSO"/>

<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"

Location="https://idp0.shib.idmgt.contoso.com/idp/profile/SAML2/POST-SimpleSign/SSO"/>

<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"

Location="https://idp0.shib.idmgt.contoso.com/idp/profile/SAML2/Redirect/SSO"/>

</IDPSSODescriptor>

<AttributeAuthorityDescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:1.1:protocol

urn:oasis:names:tc:SAML:2.0:protocol">

<Extensions>

<shibmd:Scope regexp="false">

shib.idmgt.contoso.com

</shibmd:Scope>

</Extensions>

<KeyDescriptor>

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>

MIIFpzCCBI+gAwIBAgIKa2DcuwABAAAM+jANBgkqhkiG9w0BAQUFADCBgDETMBEG

CgmSJomT8ixkARkWA2NvbTEZMBcGCgmSJomT8ixkARkWCW1pY3Jvc29mdDEUMBIG

CgmSJomT8ixkARkWBGNvcnAxFzAVBgoJkiaJk/IsZAEZFgdyZWRtb25kMR8wHQYD

VQQDExZNU0lUIE1hY2hpbmUgQXV0aCBDQSAyMB4XDTEyMTAxMTE5MzE0M1oXDTE0

MTAxMTE5MzE0M1owJTEjMCEGA1UEAxMaaWRwMC5zaGliLmlkbWd0LmFyY2hpbXMu

ZnIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCQOl+uVBg+9WqjZljy

HHIaKs1ptpMVB/dkElLaVI8/v+FuqZWeQNTH5OsQxi8XUrynsZA+9x8/r/ZFBqSu

2rK8IAZsFq4WW2uxSza+k4FaJMMShyS5rfdlxq/hnwwsmr8anJoD72xLVlygEbnu

rgzIj2wwqYfg0rKw6D1hEYTkVTqJwHNIvMQXitMVA8nlMkRMLAQUuSBmnwgyvB8G

7G4js5qi8JGrGOKHEfTW3CDE5xWUI7+3Rq6s+4F2mvPAgoXDuZITfHY5E8skRdYo

xS71yiblHcg2OIT3Vfe5iXCBOcvgQ8pv+bxEbQRe/C8CVEPIoasRsVNFYSDHudJT

2+x/AgMBAAGjggJ7MIICdzAdBgNVHQ4EFgQU8K9spMJWL6k2wdoi0PXN8MdWwkMw

CwYDVR0PBAQDAgSwMB8GA1UdIwQYMBaAFOvbEV74CZ7Y1mKc/WKd44RKKOEnMIHu

BgNVHR8EgeYwgeMwgeCggd2ggdqGT2h0dHA6Ly9tc2NybC5taWNyb3NvZnQuY29t

L3BraS9tc2NvcnAvY3JsL01TSVQlMjBNYWNoaW5lJTIwQXV0aCUyMENBJTIwMigx

KS5jcmyGTWh0dHA6Ly9jcmwubWljcm9zb2Z0LmNvbS9wa2kvbXNjb3JwL2NybC9N

U0lUJTIwTWFjaGluZSUyMEF1dGglMjBDQSUyMDIoMSkuY3JshjhodHRwOi8vY29y

cHBraS9jcmwvTVNJVCUyME1hY2hpbmUlMjBBdXRoJTIwQ0ElMjAyKDEpLmNybDCB

rQYIKwYBBQUHAQEEgaAwgZ0wVQYIKwYBBQUHMAKGSWh0dHA6Ly93d3cubWljcm9z

b2Z0LmNvbS9wa2kvbXNjb3JwL01TSVQlMjBNYWNoaW5lJTIwQXV0aCUyMENBJTIw

MigxKS5jcnQwRAYIKwYBBQUHMAKGOGh0dHA6Ly9jb3JwcGtpL2FpYS9NU0lUJTIw

TWFjaGluZSUyMEF1dGglMjBDQSUyMDIoMSkuY3J0MD8GCSsGAQQBgjcVBwQyMDAG

KCsGAQQBgjcVCIPPiU2t8gKFoZ8MgvrKfYHh+3SBT4PC7YUIjqnShWMCAWQCAQow

HQYDVR0lBBYwFAYIKwYBBQUHAwIGCCsGAQUFBwMBMCcGCSsGAQQBgjcVCgQaMBgw

CgYIKwYBBQUHAwIwCgYIKwYBBQUHAwEwDQYJKoZIhvcNAQEFBQADggEBAHrGC1bh

ajRwByTdibBK7uDIKphDIdb6ReWkBZ0XckGSRNu447gjfNoTA6KOldvHj+Qv03IV

hVjQNtLmD5NTlE3qmiDJ2Rt23k1JQ5ixZAuJRXw8Xa7Muqx5Lxov2N+CoNw9ZPpz

fXQXOL3TiWE4KiLMlvcNSUMFxmO0P3gwpFxj8MrOb5WATHOxbqItqQejqPZCfbqI

Penb+7uL1HNrbPlpyawZpB8IyLqITlkn4I8ihk75aV6mtWjNinnP2umjXYg+7x8e

lIx/JgddkWNWNTsQm4XtVbCnjunyp68SAX6OFoOBHTI2uhwYx7EB8c02ltOlsXxz

gbKSVzXZ2oam1ac=

</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</KeyDescriptor>

<AttributeService Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"

Location="https://idp0.shib.idmgt.contoso.com:8443/idp/profile/SAML1/SOAP/AttributeQuery"/>

<AttributeService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"

Location="https://idp0.shib.idmgt.contoso.com:8443/idp/profile/SAML2/SOAP/AttributeQuery"/>

<NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>

<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>

</AttributeAuthorityDescriptor>

</EntityDescriptor>

The first role descriptor *<IDPSSODescriptor>* element corresponds to the identity provider role of the Shibboleth IdP in which we are interested. As previously stated, you can see in the associated *protocolSupportEnumeration* parameter that the Shibboleth IdP supports the SAML 2.0 standard along with the SAML 1.1 standard. You can also see in the set of *<SingleSignOnService>* elements some of the URLs we’ve specified in the **Set-MsolDomainAuthentication** cmdlet.

Furthermore, modifications made to the *attribute-resolver.xml* and *attribute-filter.xml* files enable the Shibboleth IdP to issue to the Azure AD relying party SAML 2.0 assertions, which contains claims sourced from the on-premises LDAP AD LDS data source.

These claims allow the Azure AD to match the user to a shadow identity (foreign user principal) in the cloud. They are as follows:

1. The *uid* attribute of the user in the AD LDS data source as a *<saml2:NameID>* element of the subject of the SAML 2.0 assertion to issue to Azure AD in accordance of one of the *<NameIDFormat>* elements in the Azure AD metadata.

This corresponds to the unique identifier of the user tied to provisioning of the user in the Azure AD tenant of the organization, i.e. the Azure AD *ImmutableID* attribute.

This must be a unique, rename-safe identifier for the user, which must remain constant for the lifetime of the object in the cloud. This otherwise could lead to duplicate objects and unexpected synchronization errors (see section § Setting up directory synchronization);

1. The *mail* attribute of the user in the AD LDS data source as a *IDPEmail* attribute (*<saml2:Attribute>* element) of the SAML 2.0 assertion to issue to Azure AD.

The *mail* attribute of the user is also tied to the provisioned value for the user, i.e. the Azure AD *UserId* attribute.

Considering the rest of the settings in the two above files, the resulting SAML 2.0 assertion looks as follows:

<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"

ID="\_bbe9366ec1870c87b48dd975086d4574"

IssueInstant="2014-10-22T11:07:13.169Z"

Version="2.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

<saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">

https://idp0.shib.idmgt.contoso.com/idp/shibboleth

</saml2:Issuer>

<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

<ds:SignedInfo>

<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>

<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>

<ds:Reference URI="#\_bbe9366ec1870c87b48dd975086d4574">

<ds:Transforms>

<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>

<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">

<ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"

PrefixList="xs"/>

</ds:Transform>

</ds:Transforms>

<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

<ds:DigestValue>viDO5c5wDTb6CGZM+u5/cifnGmQ=</ds:DigestValue>

</ds:Reference>

</ds:SignedInfo>

<ds:SignatureValue>

c/zdBO4m2vW6MFJu8sYD+oeQ9ZEtiipdebAYG2UMkvtJlneZMinFbt+CYFZMLYUdLFE2H+zeVdIAqfLqjb

efvoGk4vG2y2XuIRYeRmY8PClblspcLK4ggzaxF8F0zU5GSlDnadVJwXFF9ipVOzo+hGnx0PpaiDP9IfNB

OwTqVCiBM9eLT2Rh7NpgF2CT+Um5x910oxdwWkGir+bR5dkHBaIbsJJmMzNfhqaSpnyLKP4ID10KLCdlO5

Gx8WnNZsEFIFIxLhyLmlEongKu2F3TED+1aAltY+pPJilwgTrqJT4ISVsQK7zY5tQyHxczlVT+mqXvz8Yn

KcjItFxgWILvjQ==

</ds:SignatureValue>

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>

MIIFpzCCBI+gAwIBAgIKa2DcuwABAAAM+jANBgkqhkiG9w0BAQUFADCBgDETMBEGCgmSJomT8ixk

ARkWA2NvbTEZMBcGCgmSJomT8ixkARkWCW1pY3Jvc29mdDEUMBIGCgmSJomT8ixkARkWBGNvcnAx

FzAVBgoJkiaJk/IsZAEZFgdyZWRtb25kMR8wHQYDVQQDExZNU0lUIE1hY2hpbmUgQXV0aCBDQSAy

MB4XDTEyMTAxMTE5MzE0M1oXDTE0MTAxMTE5MzE0M1owJTEjMCEGA1UEAxMaaWRwMC5zaGliLmlk

bWd0LmFyY2hpbXMuZnIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCQOl+uVBg+9Wqj

ZljyHHIaKs1ptpMVB/dkElLaVI8/v+FuqZWeQNTH5OsQxi8XUrynsZA+9x8/r/ZFBqSu2rK8IAZs

Fq4WW2uxSza+k4FaJMMShyS5rfdlxq/hnwwsmr8anJoD72xLVlygEbnurgzIj2wwqYfg0rKw6D1h

EYTkVTqJwHNIvMQXitMVA8nlMkRMLAQUuSBmnwgyvB8G7G4js5qi8JGrGOKHEfTW3CDE5xWUI7+3

Rq6s+4F2mvPAgoXDuZITfHY5E8skRdYoxS71yiblHcg2OIT3Vfe5iXCBOcvgQ8pv+bxEbQRe/C8C

VEPIoasRsVNFYSDHudJT2+x/AgMBAAGjggJ7MIICdzAdBgNVHQ4EFgQU8K9spMJWL6k2wdoi0PXN

8MdWwkMwCwYDVR0PBAQDAgSwMB8GA1UdIwQYMBaAFOvbEV74CZ7Y1mKc/WKd44RKKOEnMIHuBgNV

HR8EgeYwgeMwgeCggd2ggdqGT2h0dHA6Ly9tc2NybC5taWNyb3NvZnQuY29tL3BraS9tc2NvcnAv

Y3JsL01TSVQlMjBNYWNoaW5lJTIwQXV0aCUyMENBJTIwMigxKS5jcmyGTWh0dHA6Ly9jcmwubWlj

cm9zb2Z0LmNvbS9wa2kvbXNjb3JwL2NybC9NU0lUJTIwTWFjaGluZSUyMEF1dGglMjBDQSUyMDIo

MSkuY3JshjhodHRwOi8vY29ycHBraS9jcmwvTVNJVCUyME1hY2hpbmUlMjBBdXRoJTIwQ0ElMjAy

KDEpLmNybDCBrQYIKwYBBQUHAQEEgaAwgZ0wVQYIKwYBBQUHMAKGSWh0dHA6Ly93d3cubWljcm9z

b2Z0LmNvbS9wa2kvbXNjb3JwL01TSVQlMjBNYWNoaW5lJTIwQXV0aCUyMENBJTIwMigxKS5jcnQw

RAYIKwYBBQUHMAKGOGh0dHA6Ly9jb3JwcGtpL2FpYS9NU0lUJTIwTWFjaGluZSUyMEF1dGglMjBD

QSUyMDIoMSkuY3J0MD8GCSsGAQQBgjcVBwQyMDAGKCsGAQQBgjcVCIPPiU2t8gKFoZ8MgvrKfYHh

+3SBT4PC7YUIjqnShWMCAWQCAQowHQYDVR0lBBYwFAYIKwYBBQUHAwIGCCsGAQUFBwMBMCcGCSsG

AQQBgjcVCgQaMBgwCgYIKwYBBQUHAwIwCgYIKwYBBQUHAwEwDQYJKoZIhvcNAQEFBQADggEBAHrG

C1bhajRwByTdibBK7uDIKphDIdb6ReWkBZ0XckGSRNu447gjfNoTA6KOldvHj+Qv03IVhVjQNtLm

D5NTlE3qmiDJ2Rt23k1JQ5ixZAuJRXw8Xa7Muqx5Lxov2N+CoNw9ZPpzfXQXOL3TiWE4KiLMlvcN

SUMFxmO0P3gwpFxj8MrOb5WATHOxbqItqQejqPZCfbqIPenb+7uL1HNrbPlpyawZpB8IyLqITlkn

4I8ihk75aV6mtWjNinnP2umjXYg+7x8elIx/JgddkWNWNTsQm4XtVbCnjunyp68SAX6OFoOBHTI2

uhwYx7EB8c02ltOlsXxzgbKSVzXZ2oam1ac=

</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</ds:Signature>

<saml2:Subject>

<saml2:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"

NameQualifier="https://idp0.shib.idmgt.contoso.com/idp/shibboleth"

SPNameQualifier="urn:federation:MicrosoftOnline">

81372

</saml2:NameID>

<saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">

<saml2:SubjectConfirmationData Address="10.190.30.162"

InResponseTo="\_64cc8b90-7cb0-46ea-a90e-c95d4de25a80"

NotOnOrAfter="2014-10-22T11:12:13.169Z"

Recipient="https://login.microsoftonline.com/login.srf"/>

</saml2:SubjectConfirmation>

</saml2:Subject>

<saml2:Conditions NotBefore="2014-10-22T11:07:13.169Z"

NotOnOrAfter="2014-10-22T11:12:13.169Z">

<saml2:AudienceRestriction>

<saml2:Audience>urn:federation:MicrosoftOnline</saml2:Audience>

</saml2:AudienceRestriction>

</saml2:Conditions>

<saml2:AuthnStatement AuthnInstant="2014-10-22T11:07:13.091Z"

SessionIndex="9b1cabbaaea970eec250ac3beca1677d31e81c3e63a49819d91fa27a5faf84fb">

<saml2:SubjectLocality Address="10.190.30.162"/>

<saml2:AuthnContext>

<saml2:AuthnContextClassRef>

urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport

</saml2:AuthnContextClassRef>

</saml2:AuthnContext>

</saml2:AuthnStatement>

<saml2:AttributeStatement>

<saml2:Attribute FriendlyName="eduPersonScopedAffiliation"

Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9"

NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">

<saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">

member@shib.idmgt.contoso.com

</saml2:AttributeValue>

</saml2:Attribute>

<saml2:Attribute FriendlyName="UserId"

Name="IDPEmail"

NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">

<saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:type="xs:string">

user1@shib.idmgt.contoso.com

</saml2:AttributeValue>

</saml2:Attribute>

<saml2:Attribute FriendlyName="eduPersonTargetedID"

Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"

NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">

<saml2:AttributeValue>

<saml2:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"

NameQualifier="https://idp0.shib.idmgt.contoso.com/idp/shibboleth"

SPNameQualifier="urn:federation:MicrosoftOnline">

2mg/01B6ROh9eUPY5OQuriJBH7g=

</saml2:NameID>

</saml2:AttributeValue>

</saml2:Attribute>

</saml2:AttributeStatement>

</saml2:Assertion>

This XML-based signed assertion is a so-called "bearer" token, a short-live bearer token, i.e. without a proof of possession that is issued by the Shibboleth IdP to Azure AD.

Such a logon token includes a subject, an attribute statement and an authentication statement:

* **Subject**. It asserts the security principal, i.e. the subject, identified here by a *NameID* (ImmutableID);
* **Authentication statement**. It asserts that the security principal, i.e. the subject, has been authenticated by the Shibboleth IdP at a particular time using a particular method of authentication;
* **Attribute statement**. It asserts that the subject identified here by a *NameID* (ImmutableID) is associated with certain claims (*IDPEmail* in our context). Claims are provided as a string name-value pair.

The general syntax and semantics of SAML 2.0 assertions are defined in the [Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)[[23]](#footnote-23) (SAMLCore) core specification of the OASIS SAML 2.0 standard.

Finally, one should note that the Issuer URI, for example *https://idp0.shib.idmgt.contoso.com/idp/shibboleth*, in the above assertion is used to locate the namespace in Azure AD.

## Understanding the authentication flows (REVISION OF THIS SECTION IS UNDERWAY)

It’s time to consider the related authentication flows.

### Passive/web profile authentication flow

As previously noticed, the passive/web profile authentication flow is based on the web Browser SSO Profile described in the [Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)[[24]](#footnote-24) (SAMLProf) document.

In the schema below, the SAML Requester is the Azure AD and the SAML Responder the Shibboleth IdP. The User Agent is a browser.

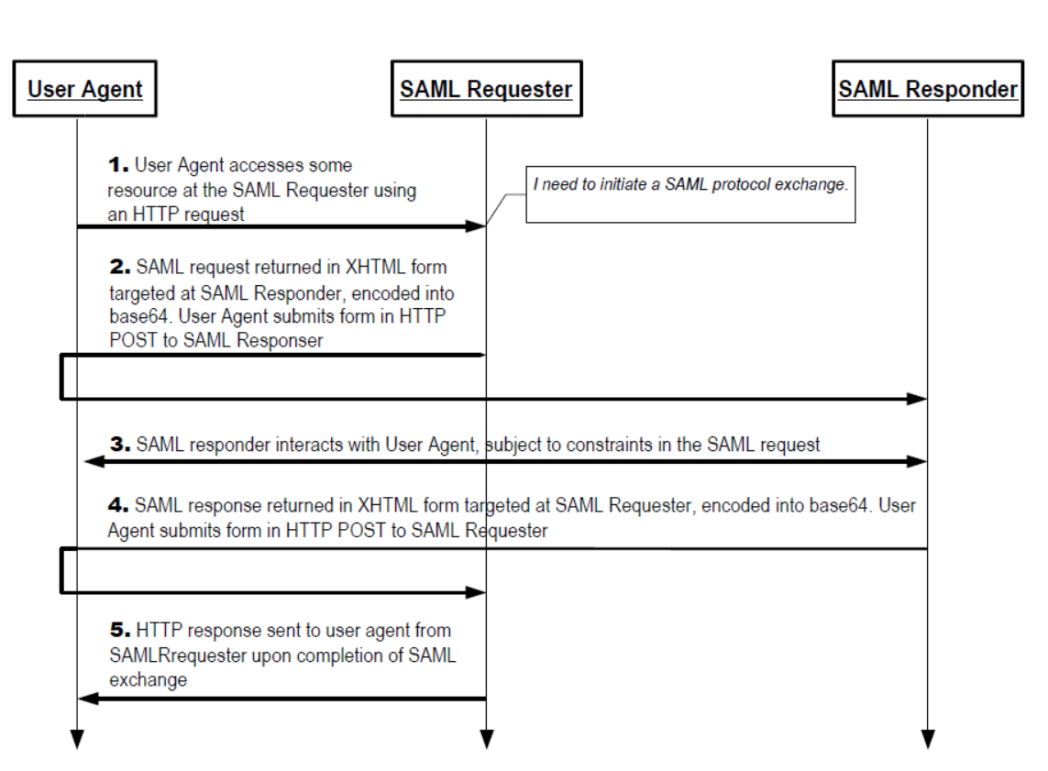


Figure 5 Passive/web profile authentication flow

This profile supports 12 possible deployment models (see section § Interaction principles and associated profiles). The one implemented here is the “SP-initiated SSO: Redirect/POST Bindings” or the HTTP-POST binding specified in the [Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)[[25]](#footnote-25) (SAMLBind) document.

It describes an SP-initiated SSO exchange where the HTTP Redirect Binding is used to deliver the *<AuthnRequest>* message to the Shibboleth IdP.

The *<AuthnRequest>* message is the request generated by Azure AD for the federated domain to initiate the authentication request, which is through an HTTPS XHTML form post to the Shibboleth IdP by the user agent.

The form contains two variables: *RelayState* and *SAMLRequest*, which is base64 encoded. The content is decoded as follows:

<?xml version="1.0" encoding="UTF-8"?>

<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"

AssertionConsumerServiceIndex="0"

ID="\_64cc8b90-7cb0-46ea-a90e-c95d4de25a80"

IssueInstant="2014-10-22T11:07:02Z"

Version="2.0"

xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">

<saml:Issuer>urn:federation:MicrosoftOnline</saml:Issuer>

<samlp:NameIDPolicy Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"/>

</samlp:AuthnRequest>

*RelayState* is HMAC(*wctx*), where *wctx* is the existing request context which contains the service request context and user login options. *wctx* is also stored in the HTTPS session cookie PPQS. The RelayState from the *<Response>* message returned by the IdP against the PPQS content.

The *AssertionConsumerServiceIndex* parameter of the <*samlp:AuthnRequest*> refers to the binding to use (HTTP-Post=0, HTTP-Post SimpleSign=1).

The HTTP POST Binding is used to return the *<Response>* message containing the SAML 2.0 assertion to Azure AD.

The *<Response>* message is posted back by the Shibboleth IdP to Azure AD’s entry point *login.srf*.

The HTTPS post contains both the form variables *RelayState* and *SAMLResponse*, which is base64 encoded. The following content is an example of the decoded SAMLResponse:

<?xml version="1.0" encoding="UTF-8"?>

<saml2p:Response xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"

Destination="https://login.microsoftonline.com/login.srf"

ID="\_8a7ec873069291d9b29795b44e47ee44"

InResponseTo="\_64cc8b90-7cb0-46ea-a90e-c95d4de25a80"

IssueInstant="2014-10-22T11:07:13.169Z"

Version="2.0">

<saml2:Issuer xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"

Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">

https://idp0.shib.idmgt.contoso.com/idp/shibboleth

</saml2:Issuer>

<saml2p:Status>

<saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>

</saml2p:Status>

<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"

ID="\_bbe9366ec1870c87b48dd975086d4574"

IssueInstant="2014-10-22T11:07:13.169Z"

Version="2.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

…

</saml2:Assertion>

</saml2p:Response>

The creation of a cookie confirms a successful login.

### Modern authentication flows

As earlier mentioned, the Office 2013/2016 client modern authentication is a new authentication stack used by Office 2013 client applications (with the March 2015 or later update) and Office 2016 client applications against Office 365. This stack allows the Office 2013/2016 client applications to engage in browser-based authentication with a Shibboleth IdP.

**Note** For more information on this stack, see the blog post [Office 2013 updated authentication enabling Multi-Factor Authentication and SAML identity providers](http://blogs.office.com/2014/11/12/office-2013-updated-authentication-enabling-multi-factor-authentication-saml-identity-providers/)**[[26]](#footnote-26)**.

Prior to ADAL based authentication, the Office 365 ProPlus/Office 2013/2016 client applications sign in flow (using the Microsoft Online Sign-In Assistant) required the WS-Trust protocol for users to sign in. Since Shibboleth IdP, does not also support WS-Trust, this prevented federated users from signing in to their Office 365 ProPlus/Office 2013/2016 Windows client applications. (This situation is shared by most of the IdPs that use the SAML 2.0 protocol, AD FS being one of the noticeable exceptions to that.) With the ADAL based authentication flows, users can sign in to the aforementioned Office Windows client applications even when using the SAML 2.0 protocols.

This diagram shows how the updated Office 365 ProPlus/Office 2013/2016 client applications enables user sign in.



Figure 7 Modern authentication flow

These Office 2013/2016 client applications use the ADAL based stack to facilitate sign in with Azure AD. When such an application makes a request to an Office 365 service, the targeted service instructs the application to visit Azure AD which speaks a simple standards based protocol: [OAuth 2.0](http://tools.ietf.org/html/draft-ietf-oauth-v2)[[27]](#footnote-27) [[28]](#footnote-28). Azure AD hosts for that purpose a web page where the user can sign in.

The Office 2013/2016 client application instructs ADAL to launch web browser control. The federation magic happens transparent to the application. The IdP could be Azure AD or a federated identity provider like Shibboleth IdP or another identity provider that uses the SAML protocol.

If the user is a federated user as covered in this document, Azure AD redirects the user to the sign in web page hosted by the Shibboleth IdP of record for the tenant. As already illustrated, this Shibboleth IdP is determined based on the domain as specified in the user’s sign in name.

The sign in web page is shown to the user on their device and the user signs in. In the context of this paper, the Shibboleth IdP returns a SAML 2.0 token to Azure AD when the user is successfully signed in. Azure AD returns a JWT token to the Office Windows client application.

**Note** JWT is a compact token format (JavaScript Object Notation (JSON) Web Token). It describes a way of representing information about the user in the token that is especially apt for REST-based software. JWT use is growing, and products supporting the format are increasingly common in the industry. For additional information, see the [eponym specification](http://tools.ietf.org/html/draft-ietf-oauth-json-web-token-30)[[29]](#footnote-29).

The Office 2013/2016 client application gets back this simple JWT token that it caches for future communication with its services. It can then use this token with services in Office 365 on behalf of the user by sending it to these services.

# Appendix A. Glossary of Shibboleth Terms

**Attribute**: a piece of information about a user. Each attribute has a unique ID and has zero of more values. Shibboleth IdP attributes are protocol-agnostic data structures.

**Attribute definition**: a plugin that creates a single attribute by transforming other attributes and state information. Shibboleth currently supports simple, scoping, regex, mapping, template, scripting, principal name, and principal authentication method attribute definitions.

**Attribute encoder**: a plugin that converts an attribute into a protocol specific form, like a SAML attribute. Attribute encoders are associated with an attribute through the attribute’s attribute definition.

**Attribute filter policy**: a policy containing a trigger, that indicates if the policy is active, and a set of attribute value filters.

**Attribute resolver**: a subsystem in Shibboleth IdP responsible for fetching, transforming, and associating encoders with attributes. Only attributes produced by attribute definitions leave the resolver and are available to other parts of the Shibboleth system.

**Attribute rule**: a rule, specific to an attribute that determines which values are released to a relying party. An attribute filter policy may have any number of attribute rules.

**Authentication mechanism**: a concrete mechanism used to authenticate a user. Shibboleth IdP currently supports REMOTE\_USER, username and password against LDAP and Kerberos protocol, and IP address based mechanisms.

**Authentication method**: an identifier that a relying party may use to stipulate how authentication should be performed. Authentication method identifiers correspond to a prescription of how authentication is done.

**Binding**: a description of how a SAML 2.0 message is attached to an underlying transport protocol, such as HTTP or SMTP. For example: If the message is sent over HTTP what HTTP headers need to be set, what are the URL or form parameter names, etc.

**Data connector**: a plugin that creates multiple attributes from information in data sources like Active Directory, other LDAP directory, SQL databases, etc. Shibboleth IdP supports static, LDAP, relational database, computed, and stored ID data connectors.

**Entity ID**: A unique identifier for an identity provider (IdP) or service provider (SP)‏. In Shibboleth IdP, the recommended format is a URL IdP: *https://<FQDN hostname>/idp/shibboleth, respectively* SP: *http://<FQDN hostname>/shibboleth*.

**Login handler**: a Shibboleth IdP component that correlates all supported authentication methods with currently configured authentication mechanisms. A login handler may map more than one authentication method to the same authentication mechanism.

**Metadata**: a description of the SAML 2.0 features supported by a SAML entity. Most importantly, this includes the URLs for communicating with an entity such as Azure AD. Shibboleth IdP and SP use this information to build technical trust between entities.

**Permit value rule**: A rule that determines if an attribute value is permitted to be released to a relying party.

**Principal connector**: a plugin that converts a name identifier, provided by a relying party, into the internally used *userid*.

**Policy requirement rule**: a specific requirement that must be met in order for an attribute filter policy to be in effect. An attribute filter policy may only have one requirement rule but some rules allow child rules to be declared and combined.

**Profile**: a description of how to use SAML 2.0, over a specific binding, to accomplish a specific task (e.g. web Browser Single Sign-On) in an interoperable manner. Profiles are the finest grained unit of interoperability within SAML 2.0.

**Relying party**: the SAML 2.0 peer to which the IdP is communicating. In all existing cases, the relying party of the IdP is always an SP. Some very advanced cases allow one IdP to be a relying party to another IdP.

**SAML:** Security Assertion Markup Language, an XML standard for exchange authentication and authorization data between security domains. SAML is standard set by the OASIS Security Services Technical Committee.

**SAML 2.0:** was ratified as an OASIS Standard in March 2005. Critical aspects of SAML 2.0 are covered in detail in the official documents SAML Conform, DAMLCore, SAMLBind, and SAMLProf.

**SAML attribute**: an attribute that is represented in SAML 2.0 notation. Shibboleth IdP transforms attributes into SAML attributes by a process known as encoding.

**Session state**: information about the user, currently active authentication methods, and services to which they are signed into such as Azure AD. A user’s IdP session is created the first time they authenticate but may outlive the lifetime of all authentication methods.

**Shibboleth:** is a suite of standards based, open source software packages for web single sign-on across or within organizational boundaries. It allows sites to make informed authorization decisions for individual access of protected online resources in a privacy-preserving manner.

The Shibboleth software suite implements widely used federated identity standards, principally OASIS SAML, to provide a federated single sign-on (SSO) and attribute exchange framework. Shibboleth IdP also provides extended privacy functionality allowing the browser user and their home site to control the attributes released to each application. Using Shibboleth-enabled access simplifies management of identity and permissions for organizations supporting users and applications. Shibboleth software is developed in an open and participatory environment, is freely available, and is released under the Apache Software License.

# Appendix B. Sample Shibboleth IdP v3 configuration files

These examples are the minimally viable samples to work with Azure AD Single Sign-On. Your Shibboleth IdP configurations may have additional elements.

## attribute-resolver.xml sample

<?xml version="1.0" encoding="UTF-8"?><AttributeResolver xmlns="urn:mace:shibboleth:2.0:resolver" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mace:shibboleth:2.0:resolver http://shibboleth.net/schema/idp/shibboleth-attribute-resolver.xsd"> <!-- ========================================== --> <!-- Attribute Definitions --> <!-- ========================================== --> <!-- Use AD LDS objectGUID for ImmutableID --> <AttributeDefinition id="ImmutableID" xsi:type="Simple" sourceAttributeID="objectGUID"> <Dependency ref="myLDAP" /> </AttributeDefinition> <!-- mail for Azure AD User ID --> <AttributeDefinition id="UserId" xsi:type="Simple" sourceAttributeID="userPrincipalName"> <Dependency ref="myLDAP" /> <AttributeEncoder xsi:type="SAML2String" name="IDPEmail" friendlyName="UserId" /> </AttributeDefinition> <!-- ========================================== --> <!-- Data Connectors --> <!-- ========================================== --> <DataConnector id="myLDAP" xsi:type="LDAPDirectory" ldapURL="%{idp.attribute.resolver.LDAP.ldapURL}" baseDN="%{idp.attribute.resolver.LDAP.baseDN}" principal="%{idp.attribute.resolver.LDAP.bindDN}" principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}" useStartTLS="%{idp.attribute.resolver.LDAP.useStartTLS:true}" connectTimeout="%{idp.attribute.resolver.LDAP.connectTimeout}" responseTimeout="%{idp.attribute.resolver.LDAP.responseTimeout}"> <FilterTemplate> <![CDATA[ %{idp.attribute.resolver.LDAP.searchFilter} ]]> </FilterTemplate> <LDAPProperty name="java.naming.ldap.attributes.binary" value="objectGUID" /> <ConnectionPool minPoolSize="%{idp.pool.LDAP.minSize:3}" maxPoolSize="%{idp.pool.LDAP.maxSize:10}" blockWaitTime="%{idp.pool.LDAP.blockWaitTime:PT3S}" validatePeriodically="%{idp.pool.LDAP.validatePeriodically:true}" validateTimerPeriod="%{idp.pool.LDAP.validatePeriod:PT5M}" expirationTime="%{idp.pool.LDAP.idleTime:PT10M}" failFastInitialize="%{idp.pool.LDAP.failFastInitialize:false}" /> </DataConnector></AttributeResolver>

## attribute-filter.xml sample

<?xml version="1.0" encoding="UTF-8"?><AttributeFilterPolicyGroup id="ShibbolethFilterPolicy" xmlns="urn:mace:shibboleth:2.0:afp" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mace:shibboleth:2.0:afp http://shibboleth.net/schema/idp/shibboleth-afp.xsd"> <AttributeFilterPolicy id="PolicyForWindowsAzureAD"> <PolicyRequirementRule xsi:type="Requester" value="urn:federation:MicrosoftOnline" /> <!-- Release mail as Azure AD User ID --> <AttributeRule attributeID="UserId"> <PermitValueRule xsi:type="ANY" /> </AttributeRule> <!-- Release Immutable ID to Azure AD --> <AttributeRule attributeID="ImmutableID"> <PermitValueRule xsi:type="ANY" /> </AttributeRule> </AttributeFilterPolicy></AttributeFilterPolicyGroup>

## metadata-provider.xml sample

<?xml version="1.0" encoding="UTF-8"?><!-- This file is an EXAMPLE metadata configuration file. --><MetadataProvider id="ShibbolethMetadata" xsi:type="ChainingMetadataProvider" xmlns="urn:mace:shibboleth:2.0:metadata" xmlns:resource="urn:mace:shibboleth:2.0:resource" xmlns:security="urn:mace:shibboleth:2.0:security" xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mace:shibboleth:2.0:metadata http://shibboleth.net/schema/idp/shibboleth-metadata.xsd urn:mace:shibboleth:2.0:resource http://shibboleth.net/schema/idp/shibboleth-resource.xsd urn:mace:shibboleth:2.0:security http://shibboleth.net/schema/idp/shibboleth-security.xsd urn:oasis:names:tc:SAML:2.0:metadata http://docs.oasis-open.org/security/saml/v2.0/saml-schema-metadata-2.0.xsd"> <MetadataProvider id="AAD" xsi:type="FileBackedHTTPMetadataProvider" metadataURL="https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml" backingFile="C:\Program Files (x86)\Shibboleth\IdP\metadata\AAD-FederationMetadata.xml" /> </MetadataProvider>

## relying-party.xml sample

<?xml version="1.0" encoding="UTF-8"?><beans xmlns="http://www.springframework.org/schema/beans" xmlns:context="http://www.springframework.org/schema/context" xmlns:util="http://www.springframework.org/schema/util" xmlns:p="http://www.springframework.org/schema/p" xmlns:c="http://www.springframework.org/schema/c" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd" default-init-method="initialize" default-destroy-method="destroy"> <bean id="shibboleth.UnverifiedRelyingParty" parent="RelyingParty"> <property name="profileConfigurations"> <list> <!-- <bean parent="SAML2.SSO" p:encryptAssertions="false" /> --> </list> </property> </bean> <bean id="shibboleth.DefaultRelyingParty" parent="RelyingParty"> <property name="profileConfigurations"> <list> <bean parent="Shibboleth.SSO" p:postAuthenticationFlows="attribute-release" /> <ref bean="SAML1.AttributeQuery" /> <ref bean="SAML1.ArtifactResolution" /> <bean parent="SAML2.SSO" p:postAuthenticationFlows="attribute-release" /> <ref bean="SAML2.ECP" /> <ref bean="SAML2.Logout" /> <ref bean="SAML2.AttributeQuery" /> <ref bean="SAML2.ArtifactResolution" /> <ref bean="Liberty.SSOS" /> </list> </property> </bean> <util:list id="shibboleth.RelyingPartyOverrides"> <!-- Microsoft Azure AD --> <bean parent="RelyingPartyByName" c:relyingPartyIds="urn:federation:MicrosoftOnline"> <property name="profileConfigurations"> <list> <bean parent="SAML2.SSO" p:encryptAssertions="false" /> <bean parent="SAML2.Logout" /> </list> </property> </bean> </util:list></beans>

## saml-nameid.xml sample

<?xml version="1.0" encoding="UTF-8"?><beans xmlns="http://www.springframework.org/schema/beans" xmlns:context="http://www.springframework.org/schema/context" xmlns:util="http://www.springframework.org/schema/util" xmlns:p="http://www.springframework.org/schema/p" xmlns:c="http://www.springframework.org/schema/c" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd" default-init-method="initialize" default-destroy-method="destroy"> <!-- ========================= SAML NameID Generation ========================= --> <!-- SAML 2 NameID Generation --> <util:list id="shibboleth.SAML2NameIDGenerators"> <ref bean="shibboleth.SAML2TransientGenerator" /> <!-- Uncommenting this bean requires configuration in saml-nameid.properties. --> <!-- <ref bean="shibboleth.SAML2PersistentGenerator" /> --> <!-- Persistent ID Generator for all entities except Azure AD --> <!-- <bean parent="shibboleth.SAML2PersistentGenerator"> <property name="activationCondition"> <bean parent="shibboleth.Conditions.NOT"> <constructor-arg> <bean parent="shibboleth.Conditions.RelyingPartyId" c:candidates="urn:federation:MicrosoftOnline" /> </constructor-arg> </bean> </property> </bean> --> <!-- Custom persistent ID Generator for Azure AD --> <bean parent="shibboleth.SAML2AttributeSourcedGenerator" p:format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" p:attributeSourceIds="#{ {'ImmutableID'} }"> <property name="activationCondition"> <bean parent="shibboleth.Conditions.RelyingPartyId" c:candidates="urn:federation:MicrosoftOnline" /> </property> </bean> <!-- <bean parent="shibboleth.SAML2AttributeSourcedGenerator" p:format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress" p:attributeSourceIds="#{ {'mail'} }" /> --> </util:list> <!-- SAML 1 NameIdentifier Generation --> <util:list id="shibboleth.SAML1NameIdentifierGenerators"> <ref bean="shibboleth.SAML1TransientGenerator" /> <!-- <bean parent="shibboleth.SAML1AttributeSourcedGenerator" p:format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress" p:attributeSourceIds="#{ {'mail'} }" /> --> </util:list> </beans>

1. Shibboleth IdP download site: http://shibboleth.net/downloads/identity-provider/latest/ [↑](#footnote-ref-1)
2. Eclipse Jetty: http://www.eclipse.org/jetty/ [↑](#footnote-ref-2)
3. *Configuration*: https://wiki.shibboleth.net/confluence/display/IDP30/Configuration [↑](#footnote-ref-3)
4. LDAPAuthnConfiguration: https://wiki.shibboleth.net/confluence/display/IDP30/LDAPAuthnConfiguration [↑](#footnote-ref-4)
5. LoggingConfiguration: https://wiki.shibboleth.net/confluence/display/IDP30/LoggingConfiguration [↑](#footnote-ref-5)
6. troubleshooting: https://wiki.shibboleth.net/confluence/display/IDP30/Troubleshooting [↑](#footnote-ref-6)
7. troubleshooting: https://wiki.shibboleth.net/confluence/display/IDP30/Troubleshooting [↑](#footnote-ref-7)
8. Configure Shibboleth for use with single sign-on: http://technet.microsoft.com/en-us/library/jj205463 [↑](#footnote-ref-8)
9. *MetadataConfiguration*: https://wiki.shibboleth.net/confluence/display/IDP30/MetadataConfiguration [↑](#footnote-ref-9)
10. *AttributeResolverConfiguration*: https://wiki.shibboleth.net/confluence/display/IDP30/AttributeResolverConfiguration [↑](#footnote-ref-10)
11. *AttributeFilterConfiguration*: https://wiki.shibboleth.net/confluence/display/IDP30/AttributeFilterConfiguration [↑](#footnote-ref-11)
12. *NameIDGenerationConfiguration*: <https://wiki.shibboleth.net/confluence/display/IDP30/NameIDGenerationConfiguration> [↑](#footnote-ref-12)
13. Shibboleth troubleshooting: https://wiki.shibboleth.net/confluence/display/IDP30/Troubleshooting [↑](#footnote-ref-13)
14. Azure Active Directory Module for PowerShell (MSOnline): <https://docs.microsoft.com/en-us/powershell/azure/active-directory/overview?view=azureadps-1.0> [↑](#footnote-ref-14)
15. MSOnline Cmdlet Reference: <https://docs.microsoft.com/en-us/powershell/module/msonline/?view=azureadps-1.0> [↑](#footnote-ref-15)
16. Add your domain: http://technet.microsoft.com/en-us/library/hh969247.aspx [↑](#footnote-ref-16)
17. Verify a domain at any domain name registrar: http://technet.microsoft.com/en-us/library/jj151803.aspx [↑](#footnote-ref-17)
18. Telerik Fiddler: http://www.telerik.com/fiddler [↑](#footnote-ref-18)
19. SAML tracer: https://addons.mozilla.org/en-US/firefox/addon/saml-tracer/ [↑](#footnote-ref-19)
20. AuthenticationCustomization: https://wiki.shibboleth.net/confluence/display/IDP30/AuthenticationConfiguration [↑](#footnote-ref-20)
21. troubleshooting: https://wiki.shibboleth.net/confluence/display/IDP30/Troubleshooting [↑](#footnote-ref-21)
22. Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf [↑](#footnote-ref-22)
23. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf [↑](#footnote-ref-23)
24. Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf [↑](#footnote-ref-24)
25. Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf [↑](#footnote-ref-25)
26. Office 2013 updated authentication enabling Multi-Factor Authentication and SAML identity providers: http://blogs.office.com/2014/11/12/office-2013-updated-authentication-enabling-multi-factor-authentication-saml-identity-providers/ [↑](#footnote-ref-26)
27. RFC 6749 The OAuth 2.0 Authorization Framework: http://tools.ietf.org/html/rfc6749 [↑](#footnote-ref-27)
28. RFC 6750 The OAuth 2.0 Authorization Framework: Bearer Token Usage: http://tools.ietf.org/html/rfc6750 [↑](#footnote-ref-28)
29. JSON Web Token (JWT): http://tools.ietf.org/html/draft-ietf-oauth-json-web-token-30 [↑](#footnote-ref-29)