### Installing the Shibboleth IdP v3 software

As already outlined, 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.

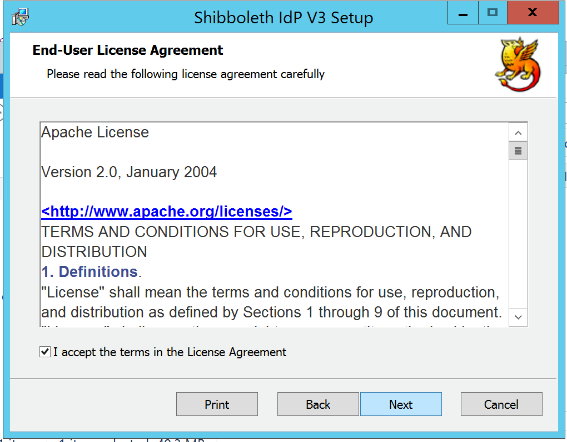
This enables to save a considerable amount of time, effort, and work at configuration time, as long as you enter the required information accurately at installation time. (Eclipse Jetty is included with the Windows installer, and Apache httpd is not required.)

To install the Shibboleth IdP software package onto the IDP0 (*shibb.zckb.me*) machine, proceed with the following steps:

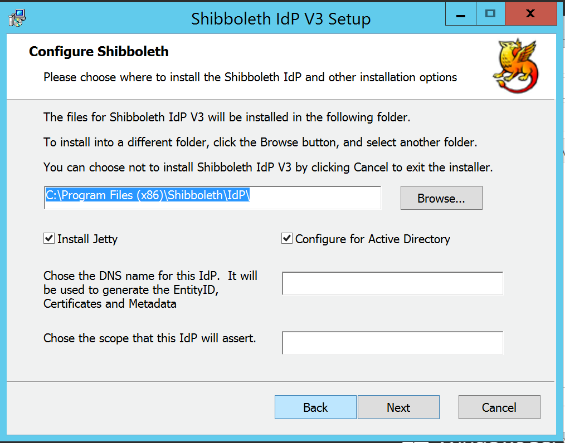
1. Visit the [Shibboleth IdP download site](http://shibboleth.net/downloads/identity-provider/latest/)[[1]](#footnote-1) and download the latest IdP (Identity Provider) software package. This guide uses the following Windows .msi installation package: shibboleth-identity-provider-3.3.2.0 -x64.msi.
2. Click **Run** to execute shibboleth-identity-provider-3.3.2.0-x64.msi. The Shibboleth IdP 3 Setup wizard opens up.



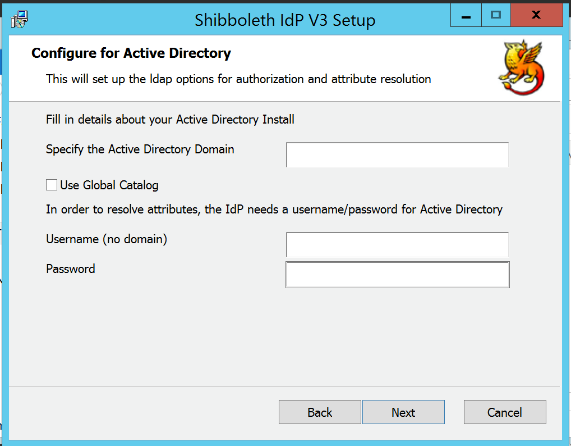
1. In the **Welcome** page, click **Next**.



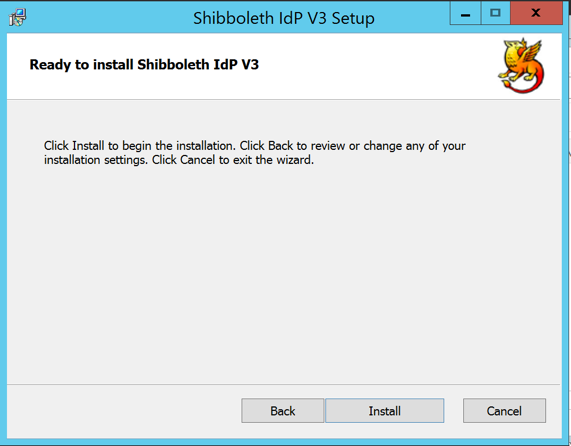
1. In the **License Agreement** page, click **Next**.



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 “*shib.zckb.me*” 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*, *shib.zckb.me*).
   6. Click **Next**.



1. In such case, this page and the next one have no suggested entries.
2. Type “*shib.zckb.me*” 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 “vmadmin” in **Username (no domain)**, and “P@ssw0rd” 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

As previously outlined, the Shibboleth IdP software must be configured once installed. The configuration typically requires to:

* 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 a number of 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 setup the trust with the Azure AD environment.

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 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=shibb, DC=zckb, DC=me**) |
| idp.authn.LDAP.useStartTLS | Whether StartTLS should be used immediately after connecting to the LDAP (Should be **false)** |
| idp.authn.LDAP.returnAttributes | List of attributes to request during authentication.  (Should be **mail,uid**) |
| 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 **vmadmin@shibb.zckb.me**) |
| idp.authn.LDAP.bindDNCredential | Search account password. (should be **P@ssw0rd**) |
| idp.authn.LDAP.dnFormat | The UPN format. (should be **%s@shibb.zckb.me**) |
| idp.attribute.resolver.LDAP.searchFilter | The filter used for attribute lookup. (should be **(sAMAccountName=$resolutionContext.principal)** ) |

**Note** Replace “**P@ssw0rd**” 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.

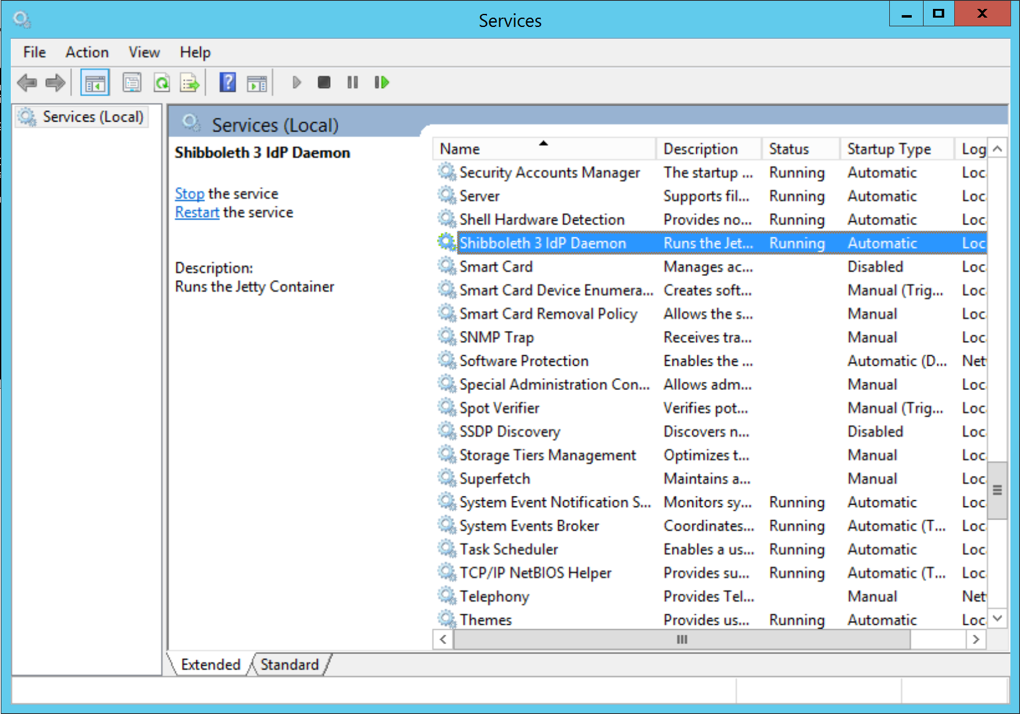
Proceed with the following steps:

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:// shibb.zckb.me/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.

Adding Azure AD as a relying party

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 the following URL:

<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)[[8]](#footnote-8) on the Shibboleth Community wiki site.

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:
4. <!-- Microsoft Azure AD Metadata -->
5. <MetadataProvider id="AAD" xsi:type="FileBackedHTTPMetadataProvider"
6. metadataURL="https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml"
7. backingFile="C:\Program Files (x86)\Shibboleth\IdP\metadata\AAD-FederationMetadata.xml" />
8. Save and close the ***metadata-provider.xml*** file.

Configure 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 list-->

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

</list>

</property>

</bean>

1. Save and close the ***relying-party.xml*** file.

Configuring the Shibboleth IdP attribute resolver

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)[[9]](#footnote-9) 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="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>

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}" responseTimeout="%{idp.attribute.resolver.LDAP.responseTimeout}"> <FilterTemplate> <![CDATA[ %{idp.attribute.resolver.LDAP.searchFilter} ]]> </FilterTemplate>

<!-- Obtain objectGUID from Windows Active Directory-->

<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)[[10]](#footnote-10) 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)[[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 ***saml-namid.xml***file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “shibboleth.SAML2NameIDGenerators”.
4. Move one line down and insert the following text to modify the list of generators that will be released:

<ref bean="shibboleth.SAML2TransientGenerator" />

     <!-- 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>

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

### Configuring the Shibboleth idP metadata

The IdP server generates a metadata file that contains configuration and integration details for SAML 2.0 single sign-on.

To configure idp metedata, proceed with the following steps:

1. Use Windows Explorer to navigate to %IDP\_Home%\metadata, e.g. C:\Program Files (x86)\Shibboleth\IdP\metadata.
2. Right-click the ***idp-metadata.xml*** file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “urn:mace:shibboleth:1.0:profiles:AuthnRequest”.
4. Move to the bottom of the file to the line before "<SingleSignOnService Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest" Location="https://shibb.zckb.me/idp/profile/Shibboleth/SSO"/>”, and insert the following text:

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

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

1. Save and close the ***idp-metadata.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)[[12]](#footnote-12) on the Shibboleth Community wiki.

### Installing Windows PowerShell for single sign-on with Shibboleth IdP

Azure AD domains are federated using Windows PowerShell and cmdlets of the Microsoft Online Services Module for Windows PowerShell.

**Note** For additional information, see the Microsoft TechNet article [Install Windows PowerShell for single sign-on with Shibboleth](http://technet.microsoft.com/en-us/library/jj205464)[[13]](#footnote-13).

**Note** Windows PowerShell is a task-based command-line shell and scripting language that is designed for system/service administration and automation. It uses administrative tasks called cmdlets. Each cmdlet has required and optional arguments, called parameters, that identify which objects to act on or control how the cmdlet performs its task. You can combine cmdlets in scripts to perform complex functions that give you more control and help you automate the administration of Windows, applications and online services in the Cloud. It has become a common way to manage the latest generation of Microsoft products and services.

For more information about Windows PowerShell 2.0, please see the [Windows PowerShell web site](http://www.microsoft.com/powershell)[[14]](#footnote-14), the [Windows PowerShell online help](http://technet.microsoft.com/en-us/library/bb978526.aspx)[[15]](#footnote-15), and the [Windows PowerShell weblog](http://blogs.msdn.com/powershell)[[16]](#footnote-16) [Windows PowerShell Software Development Kit (SDK)](http://msdn2.microsoft.com/en-us/library/aa830112.aspx)[[17]](#footnote-17) that includes a programmer’s guide along with a full reference.

To do this and to connect a Windows PowerShell command shell to Azure AD, you must have the required software for the Azure Active Directory Module for Windows PowerShell. More specifically, the local computer being used must meet the following requirements:

* Microsoft .NET Framework 3.51;
* Windows PowerShell 2.0 or above;
* Microsoft Online Services Sign-In Assistant (SIA) 7.0;

Microsoft .NET Framework 3.51 is a feature of computers running Windows Server 2008 R2, which is our configuration. Likewise, Windows PowerShell 2.0 is already installed in computers running Windows Server 2008 R2 (and above), which is our configuration.

### Installing the Microsoft Online Services Sign-In Assistant

The Microsoft Online Services Sign-In Assistant (SIA) 7.0 must be installed in order to use the Azure Active Directory Module for Windows PowerShell.

**Note** The Microsoft Online Services Sign-In Assistant (SIA) 7.0 provides end user sign-in capabilities to Microsoft Online Services, such Office 365 and Azure AD Rights Management. In the context of this paper, the SIA is used to authenticate users to these services through a set of dynamic link library files (DLLs) and a Windows service as described in the community article [Description of Microsoft Online Services Sign-In Assistant (SIA)](http://community.office365.com/en-us/w/office/534.aspx)[[18]](#footnote-18).

To install the Microsoft Online Services Sign-In Assistant (SIA) 7.0, proceed with the following steps:

1. Download the SIA package (*msoidcli\_64bit.msi*) from the following link: [Microsoft Online Services Sign-In Assistant for IT Professionals](http://go.microsoft.com/fwlink/?LinkId=286152)[[19]](#footnote-19) and click **Run** to install.



**Note** This download is intended for IT Professionals, for distribution to managed client systems as part of an Office 365 client deployment, via Microsoft System Center Configuration Manager (SCCM) or similar software distribution systems. For users who are installing Office 365 (Enterprise Preview) by means of the Office 365 Desktop Setup application, this download is not necessary, because the SIA is installed as part of the Desktop Setup process. For more information about the Office 365 desktop setup, see the Office 365 online help topic [Set up your desktop for Office 365](http://onlinehelp.microsoft.com/en-us/Office365-enterprises/ff637594.aspx)[[20]](#footnote-20).

The wizard Microsoft Online Services Sign-in Assistant Setup pops up.

1. Follow the steps of the wizard.

### Installing Azure Active Directory Module for Windows PowerShell

A Windows PowerShell "module" is a package that contains Windows PowerShell commands, cmdlets, providers, functions, variables, and aliases. The Azure Active Directory Module for Windows PowerShell is a separate installation package which includes cmdlets specifically designed for Azure AD/Office 365 administration. You run those cmdlets to set up single sign-on access to the cloud service you are subscribed to.

Administrative privileges are needed on the local computer in order to install the Azure Active Directory Module.

In order to install the tool, proceed with the following steps:

1. Navigate to the page [Set up and manage single sign-on](https://portal.office.com/IdentityFederation/IdentityFederation.aspx)[[21]](#footnote-21) on the Office 365 portal.
2. Click the download link that corresponds to the appropriate version (64-bit) of the Microsoft Online Services Module (*AdministrationConfig-en.msi*), click **Run** to execute it.

**Note** The link is also available through the Microsoft TechNet article [Install Windows PowerShell for single sign-on with AD FS](http://technet.microsoft.com/en-us/library/jj151814)[[22]](#footnote-22).



The Azure Active Directory Module for Windows PowerShell Setup wizard pops up.

1. Follow the steps of the wizard.

At this stage, the Azure Active Directory Module for Windows PowerShell installs a set of cmdlets specifically designed for Azure AD tenant-based administration.

**Note** For more information about single sign-on cmdlets, see the Microsoft TechNet articles [Use Windows PowerShell cmdlets to manage your Azure AD tenant](http://technet.microsoft.com/en-us/library/jj151805)[[23]](#footnote-23) and [Windows PowerShell cmdlet descriptions](http://technet.microsoft.com/en-us/library/jj151835.aspx)[[24]](#footnote-24).

For more information about a cmdlet that you can run in Windows PowerShell, at the Windows PowerShell command prompt, type “Get-help” and the name of the cmdlet.

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

Azure AD domains are federated using the Microsoft Online Services Module. You can use the Microsoft Online Services Module to run a series of cmdlets in the Windows PowerShell command-line interface to add or convert domains for single sign-on.

Each on-premises LDAP domain that you want to federate using Shibboleth IdP 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.

**Note** For additional information, see the Microsoft TechNet article [Set up a trust between Shibboleth and Azure AD](http://technet.microsoft.com/en-us/library/jj205457)[[25]](#footnote-25).

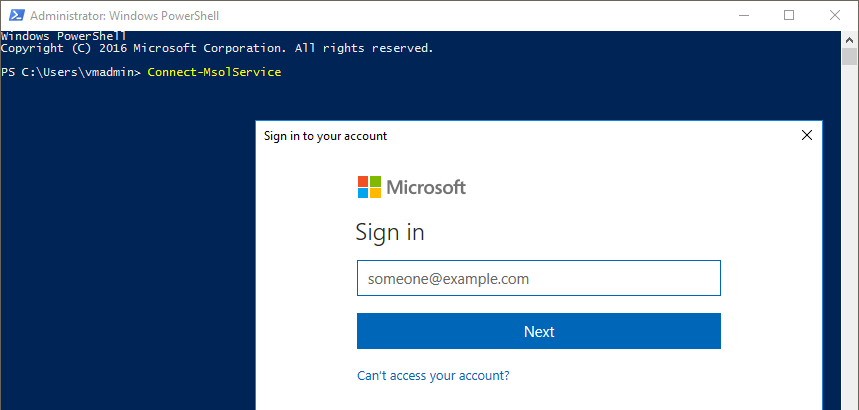
### Connecting Windows PowerShell to Azure AD

The next step is to open the Windows PowerShell from **Windows Azure Active Directory for Windows PowerShell** and connect the Windows PowerShell to the online domain using your Online Administrator Credentials.

To connect Windows PowerShell to the Azure AD, proceed as follows:

1. Open a Windows PowerShell command prompt from **Windows Azure Active Directory for Windows PowerShell.**
2. From the Windows PowerShell command prompt, type the following command:

PS C:\Windows\system32> Connect-MsolService



1. When prompted, enter the administrator account credentials of your Azure AD/Office 365 subscription.

**Note** If there is a newer version of the Windows PowerShell module, you will see a yellow warning text explaining that a newer version is available. You should always ensure that you run the latest version of the module.

*Username: eduadmin2@wwedudemo7.onmicrosoft.com*

*Password: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

### Creating the LDAP domain

We start by creating a standard domain for the LDAP *shibb.zckb.me* domain.

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

1. Connect Windows PowerShell to Azure AD (see eponym section § 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 LDAP *shibb.zckb.me* domain with the following command:

PS C:\Windows\system32> New-MsolDomain –name shibb.zckb.me

Name Status Authentication

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

shibb.zckb.me unverified Managed

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

PS C:\Windows\system32> Get-MsolDomainVerificationDns –DomainName shibb.zckb.me

canonicalName : ps.microsoftonline.com

ExtensionData : System.Runtime.Serialization.ExtensionDataObject

Capability : None

IsOptional :

Label : ms21329371.shibb.zckb.me

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.shibb.zckb.me*

*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)[[26]](#footnote-26) and [Verify a domain at any domain name registrar](http://technet.microsoft.com/en-us/library/jj151803.aspx)[[27]](#footnote-27).

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

PS C:\Windows\system32> Confirm-MsolDomain –DomainName shibb.zckb.me

### Converting the LDAP domain as a federated domain

As previously mentioned, each on-premises LDAP domain that you want to federate 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 Windows PowerShell with the **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 the services in Office 365, this supposes to convert the on-premises LDAP domain declared in the previous section, i.e. *shibb.zckb.me* in our configuration, as a federated domain.

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

1. Connect Windows PowerShell to Azure AD (see eponym section § 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, shibb.zckb.me) for single sign on:

# Connecting to Azure AD

Connect-MsolService

# Set DomainName paramater

$dom = "shibb.zckb.me"

# Set FederationBrandName paramater

$fedBrandName = "zckb Shibboleth"

# Set unique identifier of the domain in the Azure AD

$url = "https://shibb.zckb.me/idp/profile/SAML2/POST/SSO"

# Set PassiveLogOnUri parameter

$uri = "https://shibb.zckb.me/idp/shibboleth"

# Set LogOffUri parameter

$logoutUrl = "https://shibb.zckb.me/idp/profile/Logout"

# Generate new certificate using idp-signing.crt

$cert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2("C:\Program Files (x86)\Shibboleth\IdP\credentials\idp-signing.crt")

$certData = [system.convert]::tobase64string($cert.rawdata)

# Changes the authentication type of the domain.

Set-MsolDomainAuthentication -DomainName $dom -authentication managed

Set-MsolDomainAuthentication –DomainName $dom `

–federationBrandName $fedBrandName `

-Authentication Federated `

-PassiveLogOnUri $url `

-SigningCertificate $certData `

-IssuerUri $uri `

-LogOffUri $logoutUrl `

-PreferredAuthenticationProtocol SAMLP

# Gets key settings for a federated domain.

Get-MsolDomainFederationSettings -DomainName $dom

**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 Office 365 Portal. 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 Microsoft Online user accounts. The users will need to be created on premise in order 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.

### Setting up directory synchronization

At this stage, you should normally setup the directory synchronization. Considering the number of situations to take into account in AD environments (mono forest, “simple” multi-forest, “complex”, multi-forest) as well as in non-AD directories environment, and as stated in section § Non-objectives of this paper, we do not cover this subject in this paper.

**Note** Directory synchronization is not further discussed in this document. For details pertaining to this topic, please refer to [Configure directory synchronization](http://technet.microsoft.com/en-us/library/hh967629.aspx)[[28]](#footnote-28) in the Azure AD online documentation.

We instead used Windows PowerShell cmdlets hereafter to provision our test user created in section § Creating and configure a test user for the Shibboleth IdP.

This approach is not suitable for a production environment. Provisioning and synchronization are indeed not the same. Whilst with provisioning you simply create objects and/or associated resources in a directory or external system, in the case of Azure AD, the synchronization integrates the provisioning part but also enables to manage the long-term consistency/parity of state between source objects and their representation in the external system.

For the sole purpose of an illustration in this paper, you can create new Azure AD federated users from the Microsoft Online Services Module for Windows PowerShell.

To create a federated user, proceed with the following steps:

1. Connect Windows PowerShell to Azure AD (see eponym section § Connecting Windows PowerShell to Azure AD ).
2. A SAML test with Shibboleth's own tool aacli.bat:

PS C:\Program Files (x86)\Shibboleth\IdP\bin> .\aacli.bat -u=http://localhost/idp -n=testuser10 -r=urn:federation:MicrosoftOnline

{

"requester": "urn:federation:MicrosoftOnline",

"principal": "testuser10",

"attributes": [

{

"name": "ImmutableID",

"values": [

"StringAttributeValue{value=**y0rL7QDU6k+VLFeMZ9tMRg==**}"]

},

{

"name": "UserId",

"values": [

"StringAttributeValue{value=testuser10@shibb.zckb.me}"]

}

]

}

PS C:\Program Files (x86)\Shibboleth\IdP\bin>

1. Create the user with the following command:

PS C:\Windows\system32> New-MsolUser -DisplayName 'test User10' –UserPrincipalName testuser10@shibb.zckb.me -UsageLocation US -BlockCredential $false –ImmutableId **'y0rL7QDU6k+VLFeMZ9tMRg==**'

1. Also, you can re-create new user using following commands:

Remove-MsolUser -UserPrincipalName testuser10@shibb.zckb.me -Force

Remove-MsolUser -UserPrincipalName testuser10@shibb.zckb.me -RemoveFromRecycleBin

New-MsolUser -DisplayName 'Test User10' –UserPrincipalName testuser10@shibb.zckb.me -UsageLocation US -BlockCredential $false –ImmutableId '**y0rL7QDU6k+VLFeMZ9tMRg==**'

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

### Verifying single sign-on with Shibboleth IdP

As suggested in the Microsoft TechNet article [Verify single sign-on with Shibboleth](http://technet.microsoft.com/en-us/library/jj205458)[[29]](#footnote-29) 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 management 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.

To verify the SSO with the Office 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* URL which is the Identity Provider for the Microsoft Online Services.
2. Sign in using the same logon name that you use for your corporate credentials.

Username: testuser10@shibb.zckb.me

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)[[30]](#footnote-30) 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/)[[31]](#footnote-31), 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.
2. Enter the on-premises corporate credentials for the testuser10:

*Username: testuser10*

*Password: P@ssw0rd*

**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)[[32]](#footnote-32) 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)[[33]](#footnote-33) at the Shibboleth Community wiki site as well as the Microsoft TechNet article [Troubleshoot single sign-on](http://technet.microsoft.com/en-us/library/jj151834.aspx)[[34]](#footnote-34).

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. *MetadataConfiguration*: https://wiki.shibboleth.net/confluence/display/IDP30/MetadataConfiguration [↑](#footnote-ref-8)
9. *AttributeResolverConfiguration*: https://wiki.shibboleth.net/confluence/display/IDP30/AttributeResolverConfiguration [↑](#footnote-ref-9)
10. *AttributeFilterConfiguration*: https://wiki.shibboleth.net/confluence/display/IDP30/AttributeFilterConfiguration [↑](#footnote-ref-10)
11. *NameIDGenerationConfiguration*: <https://wiki.shibboleth.net/confluence/display/IDP30/NameIDGenerationConfiguration> [↑](#footnote-ref-11)
12. Shibboleth troubleshooting: https://wiki.shibboleth.net/confluence/display/IDP30/Troubleshooting [↑](#footnote-ref-12)
13. Install Windows PowerShell for single sign-on with Shibboleth: http://technet.microsoft.com/en-us/library/jj205464 [↑](#footnote-ref-13)
14. Windows PowerShell Web site: http://www.microsoft.com/powershell [↑](#footnote-ref-14)
15. Windows PowerShell online help: http://technet.microsoft.com/en-us/library/bb978526.aspx [↑](#footnote-ref-15)
16. Windows PowerShell Weblog: http://blogs.msdn.com/powershell [↑](#footnote-ref-16)
17. Windows PowerShell SDK: http://msdn2.microsoft.com/en-us/library/aa830112.aspx [↑](#footnote-ref-17)
18. Description of Microsoft Online Services Sign-In Assistant (MOS SIA): http://community.office365.com/en-us/w/office/534.aspx [↑](#footnote-ref-18)
19. Microsoft Online Services Sign-In Assistant for IT Professionals: http://www.microsoft.com/en-us/download/details.aspx?id=41950 [↑](#footnote-ref-19)
20. Set up your desktop for Office 365: http://onlinehelp.microsoft.com/en-us/Office365-enterprises/ff637594.aspx [↑](#footnote-ref-20)
21. Set up and manage single sign-on: https://portal.office.com/IdentityFederation/IdentityFederation.aspx [↑](#footnote-ref-21)
22. Install Windows PowerShell for single sign-on with AD F: http://technet.microsoft.com/en-us/library/jj151814 [↑](#footnote-ref-22)
23. Use Windows PowerShell cmdlets to manage your Azure AD tenant: http://technet.microsoft.com/en-us/library/jj151805 [↑](#footnote-ref-23)
24. Windows PowerShell cmdlet descriptions: http://technet.microsoft.com/en-us/library/jj151835.aspx [↑](#footnote-ref-24)
25. Set up a trust between Shibboleth and Azure AD: http://technet.microsoft.com/en-us/library/jj205457 [↑](#footnote-ref-25)
26. Add your domain: http://technet.microsoft.com/en-us/library/hh969247.aspx [↑](#footnote-ref-26)
27. Verify a domain at any domain name registrar: http://technet.microsoft.com/en-us/library/jj151803.aspx [↑](#footnote-ref-27)
28. Configure directory synchronization: http://technet.microsoft.com/en-us/library/hh967629.aspx [↑](#footnote-ref-28)
29. Verify single sign-on with Shibboleth: http://technet.microsoft.com/en-us/library/jj205458 [↑](#footnote-ref-29)
30. Telerik Fiddler: http://www.telerik.com/fiddler [↑](#footnote-ref-30)
31. SAML tracer: https://addons.mozilla.org/en-US/firefox/addon/saml-tracer/ [↑](#footnote-ref-31)
32. AuthenticationCustomization: https://wiki.shibboleth.net/confluence/display/IDP30/AuthenticationConfiguration [↑](#footnote-ref-32)
33. troubleshooting: https://wiki.shibboleth.net/confluence/display/IDP30/Troubleshooting [↑](#footnote-ref-33)
34. Troubleshoot single sign-on: http://technet.microsoft.com/en-us/library/jj151834.aspx [↑](#footnote-ref-34)