

Microsoft Office 365 Single Sign-On (SSO) with Shibboleth 2

Microsoft France

Published: October 2012 (updated in July 2013)

Version: 1.0a

Authors: Jean-Marie Thia (CNRS), Philippe Beraud (Microsoft France)

Contributor: Arnaud Jumelet (Microsoft France), Philippe Maurent (Microsoft Corporation)

*Copyright*

*© 2012* [*Microsoft Corporation*](http://www.microsoft.com)*. All rights reserved.*

Abstract

Through its support for the SAML 2.0 protocol, Internet2 Shibboleth 2 provides claims-based (Web) single sign-on (also known as identity federation) with the Microsoft Office 365 offering and its Web application and e-mail rich client applications (such as Outlook).

Building on existing documentation, this document is intended to provide a better understanding of the different single sign-on deployment options for Windows Azure Active Directory and the services in Office 365, how to enable single sign-on using corporate credentials and the Shibboleth 2 Identity Provider to Windows Azure Active Directory and the services in Office 365, 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 Windows Azure Active Directory/Office 365 with Shibboleth 2 along with planning and deploying such a system in their environment.

*This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.*

*Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.*

*This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.*

*© 2013 Microsoft Corporation. All rights reserved.*

*Microsoft, Active Directory, Internet Explorer, SQL Server, Windows, Windows PowerShell, and Windows Server are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners*

**Content**

[1 Introduction 1](#_Toc362296693)

[1.1 Objectives of this paper 2](#_Toc362296694)

[1.2 Non-objectives of this paper 6](#_Toc362296695)

[1.3 Organization of this paper 8](#_Toc362296696)

[1.4 About the audience 8](#_Toc362296697)

[1.5 Terminology used in this guide 9](#_Toc362296698)

[2 A brief overview of Shibboleth 2 10](#_Toc362296699)

[2.1 A short introduction on SAML 2.0 standard 12](#_Toc362296700)

[2.2 Logical architecture of the Shibboleth system components 15](#_Toc362296701)

[2.3 Interaction principles and associated profiles 18](#_Toc362296702)

[2.4 Federation metadata defined 19](#_Toc362296703)

[3 Federated authentication in Windows Azure AD/Office 365 21](#_Toc362296704)

[3.1 Sign-in Experience for Federated Identities 22](#_Toc362296705)

[3.2 Types of authentication for Federated Identities 23](#_Toc362296706)

[4 Understanding the SSO configuration and related considerations 25](#_Toc362296707)

[4.1 Preparing for the single sign-on 25](#_Toc362296708)

[4.2 Planning and deploying a Shibboleth 2 identity provider 26](#_Toc362296709)

[4.3 Configuring Shibboleth for use with single sign-on 65](#_Toc362296710)

[4.4 Installing Windows PowerShell for single sign-on with Shibboleth 2 72](#_Toc362296711)

[4.5 Setting up a trust between Shibboleth and Windows Azure AD 76](#_Toc362296712)

[4.6 Setting up directory synchronization 83](#_Toc362296713)

[4.7 Verifying single sign-on with Shibboleth 2 84](#_Toc362296714)

[4.8 Troubleshooting the single sign-on (SSO) with Shibboleth 2 92](#_Toc362296715)

[5 Understanding how federated authentication works in Office 365 93](#_Toc362296716)

[5.1 Understanding the Shibboleth 2 configuration 93](#_Toc362296717)

[5.2 Understanding the passive/Web profile authentication flow 99](#_Toc362296718)

[5.3 Understanding the ECP – Proxy Auth profile authentication flow 100](#_Toc362296719)

[Appendix A. Shibboleth 2 Glossary/Concepts 102](#_Toc362296720)

[Appendix B. Shibboleth 2 configuration file samples 104](#_Toc362296721)

# Introduction

[Microsoft Office 365](http://office365.microsoft.com/)[[1]](#footnote-1) provides secure anywhere access to professional email, shared calendars, instant messaging (IM), video conferencing, and document collaboration.

It represents the cloud version of the Microsoft communication and collaboration products with the latest version of the Microsoft desktop suite for businesses of all sizes. Office 365 indeed includes:

* **Microsoft Office**: Microsoft Office Professional Plus 2010 seamlessly connects with Microsoft Office Web Apps for a productivity experience across PCs, mobile devices, and browsers;

**Note:**

*An appropriate device, Internet connection, and supported browser are required. Some mobile functionality requires Office Mobile 2010 which is not included in Office 2010 applications, suites, or Office Web Apps. Furthermore, there are some differences between the features of the Office Web Apps, Office Mobile 2010, and the Office 2010 applications.*

* **Microsoft Exchange Online**: Exchange Online offers cloud-based email, calendar, and contacts with the most current antivirus and anti-spam solutions. It enables access to email on virtually any mobile device and takes advantage of options for voice mail, unified messaging, and archiving;
* **Microsoft SharePoint Online**: SharePoint Online is a cloud-based service for creating sites that connect colleagues, partners, and customers using enterprise social networking and customization;
* **Microsoft Lync Online**: Lync Online offers cloud-based IM, presence, and online meeting experiences with screen sharing, voice and video conferencing.

**Note:**

*For additional information on Office 365 in addition to the content of this paper, please refer to the* [*product online documentation*](http://technet.microsoft.com/en-us/library/adfs2(WS.10).aspx)*[[2]](#footnote-2), the* [*Office 365 Deployment Guide for Enterprises*](http://www.microsoft.com/download/en/details.aspx?id=26509)*[[3]](#footnote-3), the* [*Office 365 Tech Center web site*](http://technet.microsoft.com/en-us/office365/default)*[[4]](#footnote-4), and the* [*Office 365 Community web site (blogs, forums, wikis, etc.)*](http://community.office365.com/en-us/default.aspx)*[[5]](#footnote-5).*

With the exception of Internet sites for anonymous access created with SharePoint Online, users must be authenticated when accessing services in Office 365.

## Objectives of this paper

The first time you sign up for Microsoft Office 365, you are prompted to provide details about your organization and your organization’s Internet domain name registration.

This information is then used to create a new tenant for your organization in [Windows Azure Active Directory](http://technet.microsoft.com/en-us/library/hh967619.aspx)*[[6]](#footnote-6)* (Windows Azure AD).

**Note:**

*You can create an organizational domain and user accounts using this* [*page*](http://g.microsoftonline.com/0AX00en/5)*[[7]](#footnote-7) without the need to sign up for Office 365.*

Windows Azure AD provides a cloud based store for directory data and a core set of identity services including user logon processes, authentication and federation services. Office 365 relies on these capabilities for the identity management of your subscription.

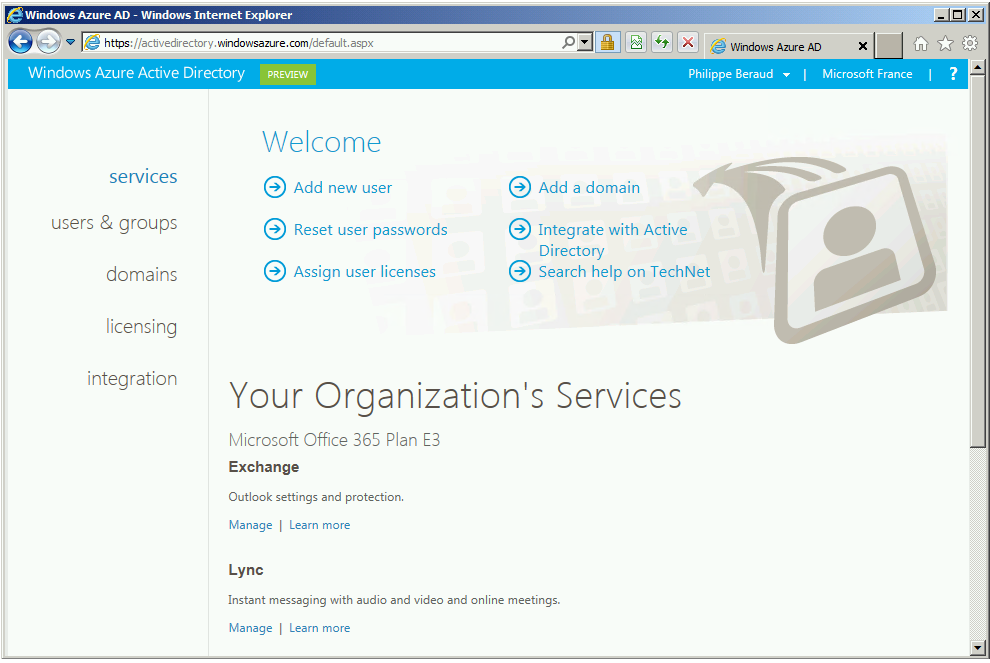
**Note:**

*Microsoft offers various Office 365 plans which are intended for different types of organizations and need. One should note that not all of them support the single sign-on (SSO) feature. This feature is supported with all of the enterprise plans. In this document, we illustrate the feature through an E3 plan subscription.*

Interestingly enough, organizations that subscribe to Office 365 can then use Windows Azure AD to configure SSO to allow interoperability with their existing on-premises identity management environment. This can be configured via either the Microsoft Online Portal (MOP) or the (preview of the) [Windows Azure AD Management Portal](https://activedirectory.windowsazure.com/)[[8]](#footnote-8).

**Note:**

*These two portals all read from and write to a single shared instance of Windows Azure AD that is associated with your organization’s tenant. In this way, the portals act as a front-end interface that pull in and/or modify your tenant data. This said, you can use the Windows Azure AD portal to do most of the functions, with the added benefit of having a single place to see all of the users, groups, domains, licenses associated with all of the cloud services that your organization subscribes to.*



By leveraging the SSO feature of the Windows Azure AD organization’s tenant, Office 365 provides in turn organizations with the ability to authenticate against the organization’s on-premises identity infrastructure (such as Active Directory or LDAP), allowing their users to use their corporate credentials to access their services in Office 365 that they have been provisioned for.

Thus, users that are on the internal corporate network or connected through a VPN will have seamless access to services in Office 365. If users are accessing services in Office 365 from home or from any computer not connected to the corporate network, they will also still have access to services in Office 365 using their corporate credentials.

Such a user sign-in experience is awaited by many of the organizations:

* **Work computer on a corporate network**: When users are at work and signed in to the corporate network, SSO enables them to access the services in Office 365 (without signing in again, depending on the on-premises security token services (STS) being used to sustain this feature);
* **Home or public computer**: the users must sign in with corporate credentials to access the services in Office 365. This is still an advantage since they will only have to remember one set of credentials for their corporate and Office 365 accesses;
* **Smartphone**: on a smartphone, in order to access Microsoft Exchange Online using Microsoft Exchange ActiveSync (EAS), the users must sign in with their corporate credentials. This is still an advantage since they will only have to remember one set of credentials for their corporate and Office 365 accesses;
* **Microsoft outlook or other e-mail clients**: the users must sign in with their corporate credentials to access their e-mail messages if they are using Outlook or an e-mail client that is not part of Office such as an IMAP or a POP client.

**Note:**

*You only need to sign up for a Windows Azure AD tenant one time and then you can sign-in to that same tenant when you want to subscribe to other Microsoft cloud services like* [*Windows Intune*](http://technet.microsoft.com/en-us/windows/intune.aspx)*[[9]](#footnote-9) for instance. By using your organization’s Windows Azure AD tenant in this way, any additional services that you might decide to subscribe to in the future can fully leverage the existing user accounts, policies, settings or on-premises server deployments you may have already configured previously to help improve efficiencies between your organization’s identity infrastructure on-premises and Windows Azure AD.*

*For instance, if you originally signed up for an Office 365 subscription and previously went through the steps to further integrate your on-premises identity infrastructure with your Windows Azure AD tenant by configuring directory synchronization and/or single sign-on, you can later sign up for another Microsoft cloud service such as Windows Intune which can also leverage the precise same directory integration benefits as Office 365.*

As of writing, the primary option in terms of STS for the SSO feature of Windows Azure AD is Active Directory Federation Services (AD FS) 2.0 service that the organization deploys on-premises and then configures Windows Azure AD to securely communicate.

AD FS 2.0 is a component of the Windows (Server) platform and, as such, the right to use it is included in the associated license costs.

**Important note:**

*The AD FS role available in Windows Server 2008 (R2) doesn’t correspond to AD FS 2.0; this is the previous version 1.1 instead. The AD FS 2.0 software package for your specific operating system version (either Windows Server 2008 or Windows Server 2008 R2) is the AdfsSetup.exe setup file. To download this file, go to* [*Active Directory Federation Services 2.0 RTW*](http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=10909)*[[10]](#footnote-10).*

**Important note:**

*As of writing, update rollup 2 for AD FS 2.0 is available. This update rollup (or the previous one[[11]](#footnote-11)) includes hotfixes and updates for AD FS 2.0 RTW that are of special interest in the context of this paper for the single sign-on feature of Office 365. For more information about this update rollup and its download, please see article 2681584* [*Description of Update Rollup 2 for Active Directory Federation Services (AD FS) 2.0*](http://support.microsoft.com/kb/2681584)*[[12]](#footnote-12).*

For more information on planning and configuring SSO for Office 365 with AD FS 2.0, please refer to the white paper entitled [Microsoft Office 365 Single Sign-On (SSO) with AD FS 2.0](http://www.microsoft.com/en-us/download/details.aspx?id=28971)[[13]](#footnote-13) available on the Microsoft Download Center.

It aims at providing a better understanding of the different single sign-on deployment options for Windows Azure AD and the services in Office 365 with AD FS 2.0, how to enable single sign-on using corporate Active Directory (AD) credentials and AD FS 2.0 to the services in Office 365, and the different configuration elements to be aware of for such deployment.

Special thanks to Ross Adams, Microsoft Senior Program Manager, for the provided valuable content on this subject such as the MSDN Channel 9 webcast [Microsoft Office 365: Identity   
and Access Solutions](http://channel9.msdn.com/Events/TechEd/NorthAmerica/2011/OSP215)[[14]](#footnote-14).

In addition to AD FS 2.0, support for additional claims/identity providers, e.g. Shibboleth 2 (as well as [Optimal IDM Virtual Identity Server Federation Services](http://go.microsoft.com/fwlink/?LinkID=266318)[[15]](#footnote-15) and [PingFederate 6.10](http://go.microsoft.com/fwlink/?LinkID=266320)[[16]](#footnote-16)) has been added, so that organizations may continue to use their existing on-premises identity infrastructure for single sign-on with Windows Azure AD and the Microsoft Online services such as Office 365, whether this identity infrastructure is based on AD or on non-AD directories. For organizations that already have a federated SSO infrastructure in place, it is indeed highly desirable to reuse it for Office 365.

As of today, the Shibboleth 2 Identity Management system is notably leveraged by many educational and research institutions around the world. Shibboleth 2 supports different data sources (Active Directory, other LDAP directory, SQL databases, etc.) that can be used as an identity repository.

Shibboleth 2 is an implementation of identity provider using OASIS standard [Security Assertion Markup Language (SAML) 2.0](http://go.microsoft.com/fwlink/?LinkId=193996)[[17]](#footnote-17) protocol to provide Web single sign-on across or within organizational boundaries

Shibboleth 2 provides cross-domain (Web) SSO (also known as identity federation) with Microsoft and non-Microsoft federation solutions.

[Wikipedia](http://en.wikipedia.org/wiki/Federated_identity)[[18]](#footnote-18) defines identity federation as follows:

“*Federated identity, or the ‘federation’ of identity, describes the technologies, standards and use-cases which serve to enable the portability of identity information across otherwise autonomous security domains. The ultimate goal of identity federation is to enable users of one domain to securely access data or systems of another domain seamlessly, and without the need for completely redundant user administration.*”

Built on existing Internet2 and Microsoft documentation and knowledge base articles, this paper further presents the single sign-on feature (also known as identity federation) of Windows Azure AD with Shibboleth 2.

For that purpose, beyond a short depiction of the Shibboleth 2 technology to introduce key concepts, requirements, and components, it:

* Describes the different identity options in Windows Azure AD;
* Shortly depicts in this context the identity architecture and features in Windows Azure AD;
* Describes the various implementation scenarios for federated authentication with Shibboleth 2;
* Describes how federated authentication works with Shibboleth 2.

On that basis, Windows Azure AD and Microsoft Cloud services projects involving Shibboleth 2 in this context can be hopefully more easily completed, and thus consequently enabling customers to realize the full potential of these offerings.

**Beyond the Shibboleth 2 support, it should be noted that generic SAML 2.0 support is not provided.**

## Non-objectives of this paper

Whilst single sign-on is not required for directory synchronization (but it will provide a richer user experience), directory synchronization is however a requirement for single sign-on.

Hence, the implementation of directory synchronization is needed in order to keep the on-premises identities in sync with the organization’s Windows Azure AD tenant.

One of the benefits is that it enables controlling and managing the corporate user account in the traditional way through the existing tooling that accompanies the on-premises identity infrastructure.

This one piece really does provide seamless user management between the on-premises and Windows Azure AD environments.

Organizations that use Active Directory, can easily implement this through the [Microsoft Online Services Directory Synchronization Tool](http://onlinehelp.microsoft.com/en-us/office365-enterprises/ff652545.aspx) (DirSync)[[19]](#footnote-19). The DirSync tool enables service administrators keeping Windows Azure AD, i.e. Office 365, users, contacts, and groups updated with changes made in the on-premises Active Directory Domain Services (AD DS) for a single Active Directory forest.

Directory synchronization is not something that is new for Windows Azure AD. The DirSync tool is built on top of Microsoft Forefront Identity Manager (FIM) 2010. The configuration of directory synchronization has been simplified for the Windows Azure AD environment. There is no manual configuration that you need to be concerned with, everything being configured via wizards.

The initial version of the DirSync tool was intended for an on-premises Active Directory mono-forest. However, many customers are operating with more than one on-premises Active Directory forest for a variety of reasons, such as:

* Multiple account forests, where user accounts exist in multiple Active Directory forests;
* Account and resource forests, where the Exchange organization is typically hosted in a resource forest model.

 Through the above, the following scenarios can also be considered:

* Multiple account forests, with no on-premises Exchange;
* Single account forests, single exchange resource forest;
* Multiple account forest, single exchange resource forest.

Multi-forest topologies will be supported with Windows Azure AD today for customers that meet certain prerequisites, allowing synchronizing from multiple forests, and, in the context of this paper, supporting single sign-on with multiple AD forests.

For that purpose, a new version of the DirSync tool is soon going to be available, to support “simple” multi-forest configurations.

Only limited “simple” multi-forest scenarios can be supported, as categorized by the table hereafter.

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Simple** | **Complex** |
| *Do you move objects between on-premises AD forests?* | No | Yes |
| *Do you have security groups that contain objects from more than one on-premises AD forest?* | No | Yes |
| *Do you have duplicated objects across on-premises AD forests, i.e. either the same object appears in multiple forests, or some subset of attributes are mastered for a user object in one forest, another subset are mastered in another forest?* | No | Yes |
| *Do you have an Exchange resource forest, or require an Exchange hybrid deployment, aka "rich coexistence"?* | No | Yes |

**Important note:**

*Users in each on-premises AD forest require a different User Principal Name (UPN) suffix. For example @forestA.com in forest A and @forestB.com in forest B. Shared UPN namespaces across forest are not supported by design, i.e. you cannot provide a single UPN namespace for all users across multiple AD forests.*

The table above provides an overview of some of the limitations of the “simple” multi-forest DirSync tool:

* No support for multi-forest Exchange hybrid deployment features;
* No support for scenarios where users have some attributes mastered in one AD forest, and other attributes mastered in another (such as in the account/resource forest model);
* No support for user moves between on-premises forests. This would indeed result in user being deleted in the cloud THEN recreated; will result in mailbox deletion.

To cover scenarios that are impacted by these limitations, FIM 2010 and the Windows Azure AD Connector (Management Agent) for FIM 2010 (formerly known as the Office 365 Connector for FIM 2010) is required instead of the DirSync tool in order to handle the directory synchronization in a “complex” multi-forest Active Directory environment. Availability of the Windows Azure AD Connector for FIM 2010 is currently subject to certain pre-requisites and an organization must work directly with Microsoft.

**As of writing, the Windows Azure AD Connector for FIM 2010 is indeed provided via Microsoft Premier Deployment (MPD) only. Please contact your local Microsoft Office 365 CXP Solution Architect if “complex” multi-forest support is a requirement for your organization.**

For environments that rely on/include non-AD directories, FIM 2010 and the Windows Azure AD Connector for FIM 2010 are also required.

**As for multi-forest support, availability of this solution is currently subject to same pre-requisites. Please contact your local Microsoft Office 365 CXP Solution Architect if Non-AD directories are a requirement for your organization.**

It is recommended to first install and configure single sign-on, and then implement directory synchronization. This is not a hard requirement but it is recommended.

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)[[20]](#footnote-20) in the Windows Azure AD online documentation.

## Organization of this paper

To cover the aforementioned objectives, this document is organized by themes which are covered in the following sections:

* A brief overview of Shibboleth 2;
* Federated authentication in Windows Azure AD/Office 365;
* Understanding the SSO configuration and related considerations;
* Understanding how federated authentication works in Office 365.

## About the audience

(Cross-domain) SSO − also known as identity federation − in general is a broad topic, with many facets, depths of understanding, protocols, standards, tokens, etc. This paper addresses the SSO topic only from the Windows Azure AD/Office 365 perspective and from both conceptual and technical levels.

As of writing and as previously mentioned, AD FS 2.0, Shibboleth 2, Optimal IDM Virtual Identity Server Federation Services and PingFederate 6.10 are the only supported technologies to enable this capability (even if this is something that may evolve in the future).

This document is intended for system architects and IT professionals who are interested in understanding how to enable and configure this SSO capability of Windows Azure AD with Shibboleth 2 to seamlessly access the services of Office 365.

It has the exact same objectives of the aforementioned white paper [Microsoft Office 365 Single Sign-On (SSO) with AD FS 2.0](http://www.microsoft.com/en-us/download/details.aspx?id=28971) but with the Shibboleth 2 technology instead of AD FS 2.0. It is part of a series of documents on the identity and security features of Office 365.

**Note:**

*For more information on leveraging Optimal IDM Virtual Identity Server Federation Services and PingFederate 6.10 solutions as identity providers to implement single sign-on, see the online help topic* [*Use third-party identity providers to implement single sign-on*](http://technet.microsoft.com/en-us/library/jj679342.aspx)*[[21]](#footnote-21). For the PingFederate instructions on how to configure the solution to provide the single sign-on experience to your Active Directory users, see the document* [*Web SSO to Office 365 Using PingFederate*](http://go.microsoft.com/fwlink/?LinkID=266321)*[[22]](#footnote-22).*

## Terminology used in this guide

Throughout the rest of this document, there are numerous references to federation concepts that are called by different names in the Microsoft and Internet2 products and/or technologies. The following table assists in drawing parallels between the two.

|  |  |  |
| --- | --- | --- |
| **Concept** | **Microsoft name** | **Shibboleth name** |
| XML document describing a user sent during an access request from the federation party that is handling users to the federation party that is managing an application | Security Token | Assertion |
| Partner in a federation that creates security tokens for users | Claims Provider | Identity Provider (IdP) |
| Partner in a federation that consumes security tokens for providing access to applications | Relying Party (RP) | Service Provider (SP) |
| Data about users that is sent inside security tokens | Claims | Assertion attributes |

In this document, Shibboleth 2 software performs Claims Provider/Identity Provider role, Windows Azure AD both the Claims Provider/Identity Provider role and the Relying Party/Service Provider role, and finally the services in Office 365 the Relying Party/Service Provider role.

Windows Azure AD acts as gateway between Shibboleth 2 and the services in Office 365. The Windows Azure AD sign-in service (STS) indeed extends the Microsoft Federation Gateway (MFG) to support SAML 2.0 protocol apart from supporting the WS\* protocol, in order to maximize the interoperability.

# A brief overview of Shibboleth 2



This section quickly describes the Shibboleth 2 system so that it is simpler to understand how Windows Azure AD/Office 365 can be exposed in a Shibboleth identity federation from a technology perspective, as well a protocol perspective with SAML 2.0.

**Note:**

*For more information about Shibboleth 2, see the* [*Shibboleth 2 home page*](http://go.microsoft.com/fwlink/?LinkID=256497)*[[23]](#footnote-23).*

Shibboleth (as a reference to the Hebrew word "shibbóleth” and the related Biblical use, i.e. to discover hiding members of the opposing group) was designed to fill higher education needs in terms of identity federation and attributes propagation for a number of partners. The Shibboleth project was initiated by the [Internet2/MACE (Middleware Architecture Committee for Education)](http://middleware.internet2.edu/MACE)[[24]](#footnote-24) in 2000. The project now hosted by the [Shibboleth Consortium](http://shibboleth.net/)[[25]](#footnote-25) refers to both a specification and an open source project that implements them as a distributed system.

As a specification, Shibboleth is an extension of the SAML (Security Assertion Markup Language) 1.1 standard which comes from the [OASIS Security Services (SAML) TC](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security)[[26]](#footnote-26); this technical committee defined a protocol to exchange security information in order to implement Web SSO.

SAML 1.1 standard is an XML-based standard for exchanging authentication and authorization data between security domains/realms, that is, between a Service Provider (SP), a consumer of (SAML) assertions and an Identity Provider (IdP), a producer of (SAML) assertions:

1. The SP provides (or does not provide) Web resources depending on the trusted user attributes it received in the (SAML) assertions in order to feed it access control;
2. The IdP is in charge of authenticating users and providing attributes based on that authentication, which are vehicle in (SAML) assertions.

SAML 1.1 profiles assume the user starts with the IdP. In order to take into account scenarios where a user first connects to an SP, Shibboleth extended the SAML browser/POST and browser/Artifact profiles.

The Shibboleth project connected with the work of the technical committee, participating in SAML from its initiation.

For that, the Shibboleth specification:

1. Defines the notion of an authentication request which allows an SP to require some authentication from a browser
2. Introduces the notion of WAYF (*Where Are You From?*)[[27]](#footnote-27), which allows a browser to select its IdP.

In other terms, the WAYF service is an optional service that can redirect a user towards a security domain where his identity is declared as an account, and more precisely towards an IdP of this security domain.

1. Also specifies how attributes exchanges take place (this is not the case in SAML 1.1).

As an implementation, Shibboleth 1.0 was released in 2003, and was quickly adopted by the worldwide Higher Education/Research community. It implements the Shibboleth specification as SAML 1.1 extensions.

**SAML 2.0 was published as an OASIS standard in 2005.** [**Shibboleth 2**](http://go.microsoft.com/fwlink/?LinkID=256497)**[[28]](#footnote-28) was released the following year. Shibboleth 2 provides SAML 2.0 support (as well backward compatibility with Shibboleth 1.3).**

SAML 2.0 results from the convergence of the previous version of the standard itself, i.e. SAML 1.1, and from the following two extensions/specifications based on it forming the foundation for the standard:

* Internet2 Shibboleth 1.3;
* [Liberty Identity Federation Framework (ID-FF) 1.2](http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_ff_1_2_specifications/?f=resource_center/specifications/liberty_alliance_id_ff_1_2_specifications)[[29]](#footnote-29).

Like SAML 1.1, the ID-FF specification is a cross-domain, browser-based, Single Sign-On (SSO) framework. In addition, the specification defined the notion of circle of trust (CoT), where each participating domain/realm is trusted to accurately document the processes used to identify a user, the type of authentication used, and any policies associated with the resulting authentication credentials. Other members of the circle of trust may examine these policies to determine whether to trust such information. The CoT represents a static trust schema. Liberty Alliance contributed its federation specification to OASIS.



In an effort to grow the identity marketplace, Liberty Alliance also introduced the [Liberty Interoperable certification program](http://www.projectliberty.org/liberty/liberty_interoperable)[[30]](#footnote-30), operated by the [Drummond group](http://www.drummondgroup.com/)[[31]](#footnote-31), and designed to test commercial and open source products against its own specifications like the aforementioned ID-FF specification and published standards like the SAML standard to assure base levels of interoperability between products.

As of writing, over 80 solutions from numerous vendors and organizations worldwide have passed testing.

In June 2009, all Liberty Alliance work and related materials have been contributed to the [Kantara Initiative](http://kantarainitiative.org/)[[32]](#footnote-32) (kan-TAR-a: swahili for "bridge"; arabic roots in "harmony"). The project Web site remains as an archive of the work of the Liberty Alliance.



Kantara Initiative is working to “*bridge and harmonize the identity community with actions that will help ensure secure, identity-based, online interactions while preventing misuse of personal information so that networks will become privacy protecting and more natively trustworthy environments*”.

As a consequence of this transition, the SAML 2.0 interoperability certification program formerly run from Liberty Alliance is now handled by the Kantara Initiative.

## A short introduction on SAML 2.0 standard

oasis.png

The OASIS SAML 2.0 standard is a suite of specifications and, as such, comprises a set of normative and non-normative documents:

* [SAML V2.0 Executive Overview](file:///C:\Users\philber\AppData\Local\Microsoft\Windows\Temporary%20Internet%20Files\Content.Outlook\C63FN0RZ\•%09http:\www.oasis-open.org\committees\download.php\13525\sstc-saml-exec-overview-2.0-cd-01-2col.pdf)[[33]](#footnote-33) (SAMLExecOvr);
* [Security Assertion Markup Language (SAML) V2.0 Technical Overview](http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf)[[34]](#footnote-34) (SAMLTechOvw);
* [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)[[35]](#footnote-35) (SAMLCore), the core specification*;*
* [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)[[36]](#footnote-36) (SAMLBind), which maps the SAML messages onto the standard messaging or communication protocols;
* [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)[[37]](#footnote-37) (SAMLProf), the use cases or the “How-to” in regards to the use of SAML to solve specific problems of the extended enterprise;
* [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)[[38]](#footnote-38) (SAMLMeta), the configuration data (endpoint URLs, key material for verifying signatures, etc.) to establish trusts between SAML entities;
* [Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0](http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf)[[39]](#footnote-39) (SAMLAuthnCxt), a detailed description of the user authentication mechanisms;
* [Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0](http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf)[[40]](#footnote-40) (SAMLConform), the operational modes for the SAML 2.0 implementations;
* [Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0](http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf)[[41]](#footnote-41) (SAMLSec), an analysis of both the security and the privacy in SAML 2.0;
* [Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0](http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf)[[42]](#footnote-42) (SAMLGloss), the terminology used in SAML 2.0.

**Note:**

*Following the release of the SAML 2.0 standard in 2005, the OASIS SAML TC has continued work on several enhancements. These documents are available from the OASIS SAML TC Web site.*

In order to have a good understanding of the standard and be able to further dig into the nitty-gritty details if needed to, for instance solve interoperability issue, we strongly advise to start reading the non-normative SAMLTechOvw document, which, as its name indicates, gives the key to understand the standard and its organization and ramification.

The critical aspects of SAML 2.0 are covered in detail in the normative documents SAMLCore, SAMLBind, SAMLProf, and SAMLConform.

SAML 2.0 defines XML-based **assertions** and **protocols**, **bindings**, and **profiles**. The term SAML Core, in relationship with the SAMLCore core specification, refers to the general syntax and semantics of SAML assertions as well as the protocol used to request and transmit those assertions from one system entity to another. SAML assertions are usually transferred from an IdP to a SP.

In Windows Azure AD terminology, SAML 2.0 refers to a subset of SAML 2.0 protocols and bindings. This is distinct from supporting the SAML 2.0 assertions used in WS-Federation and WS-Trust login flows, though SAML protocols also use SAML assertions, and differs from AD FS 2.0, and Windows Identity Foundation (WIF) terminology where SAML refers to the tokens and SAMLP is used to refer to the protocols and bindings.

### SAML 2.0 Assertions

A SAML 2.0 assertion is a (signed) (security) token and can be seen as the vehicle/container for (security) information. Such assertions contain beyond a subject and conditions, which apply to the assertions, statements or claims that SPs typically use to make or derive access control decisions. Three types of statements are provided by SAML:

* Authentication statement, which asserts that the security principal has been authenticated by the IdP at a particular time using a particular method of authentication. An authentication context may also be disclosed as such a statement;
* Attribute statement, which asserts that a subject is associated with certain attributes. An attribute is typically a string name-value pair. Relying parties use these attributes or claims to make or derive access control decisions;
* Authorization decision statement, which asserts that a subject is allowed to perform a specific action on specific resource given specific evidence;

**Note:**

*The vocabulary is intentionally limited to promote another OASIS standard instead: the eXtensible Access Control Markup Language (XACML) governed by the eponym* [*OASIS TC*](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml)*[[43]](#footnote-43). XACML is a XML-based declarative access control policy language and a processing model describing how to interpret the policies.*

In the context of this paper, the SAML assertion we have to consider is the so-called "bearer" assertion, a short-lived bearer token (i.e. without a proof of possession) issued by an IdP to a SP. Such an assertion includes both an authentication statement and an attribute statement.

### SAML 2.0 protocols

A SAML 2.0 protocol describes how certain SAML elements (including assertions) are packaged within SAML request and response elements, and gives the processing rules that SAML entities like IdP and SP must follow when producing or consuming these elements. For the most part, a SAML protocol is a simple request-response protocol.

It is important to keep in mind that a SAML protocol always refers to what is transmitted, and not how (the latter is determined by the choice of binding).

In the context of this paper, the most interesting SAML protocols are the Authentication Request Protocol, the Artifact Resolution Protocol, and the Single Logout Protocol.

### SAML 2.0 bindings

A SAML 2.0 binding determines how SAML requests and responses map onto standard messaging or communications protocols. In other words, it’s a mapping of a SAML protocol message onto standard messaging formats and/or communications protocols. SAML 2.0 completely separates the binding concept from the underlying profile (see next section).

The SAML 2.0 standard defines several bindings:

* HTTP Redirect (GET) binding;
* HTTP POST binding;
* HTTP Artifact binding;
* Etc.

### SAML 2.0 Profiles

A SAML 2.0 profile is a concrete manifestation of a defined use case using a particular combination of assertions, protocols, and bindings, assertions. Indeed, it describes in detail how SAML 2.0 assertions, protocols, and bindings combine to support the considered use case.

The SAML 2.0 standard defines several profiles:

* Web Browser SSO profile;
* Artifact Resolution profile;
* Single Logout profile;
* Identity Provider Discovery profile;
* Enhanced Client or Proxy (ECP) profile,
* Etc.

The most important one is certainly the Web Browser SSO profile since this is the primary SAML use case for Web SSO and federation. For additional information, see section § 2.3 Interaction principles and associated profiles.

## Logical architecture of the Shibboleth system components

This description of the components as well as the interaction between them is how the authors of this document see the system provided by the Shibboleth Consortium through the [Shibboleth Service Provider](http://shibboleth.net/products/service-provider.html)[[44]](#footnote-44) and [Shibboleth Identity Provider](http://shibboleth.net/products/identity-provider.html)[[45]](#footnote-45) software.

**Note:**

*For an introduction to Shibboleth, see* [*Understanding Shibboleth*](https://wiki.shibboleth.net/confluence/display/SHIB2/UnderstandingShibboleth)*[[46]](#footnote-46)* *which describes the system and defines the Shibboleth project terminology.*

*You can also consult the* [*Shibboleth Architecture Technical Overview*](http://open-systems.ufl.edu/files/draft-mace-shibboleth-tech-overview-latest.pdf)*[[47]](#footnote-47) document. While this document is about the version 1.x of Shibboleth, the main elements are still valid in the context of Shibboleth 2, which this document is about.*

*In the context of a French speaking community, as well as for planning considerations in your specific environment, you are encouraged to read resources made available in the context of the Education & Research federation (so-called federation Education-Recherche), a service provided by “GIP RENATER”. Those resources are available online on the* [*federation Education-Recherche Web site*](https://services.renater.fr/federation/index)*[[48]](#footnote-48).*

### Shibboleth Service Provider (SP)

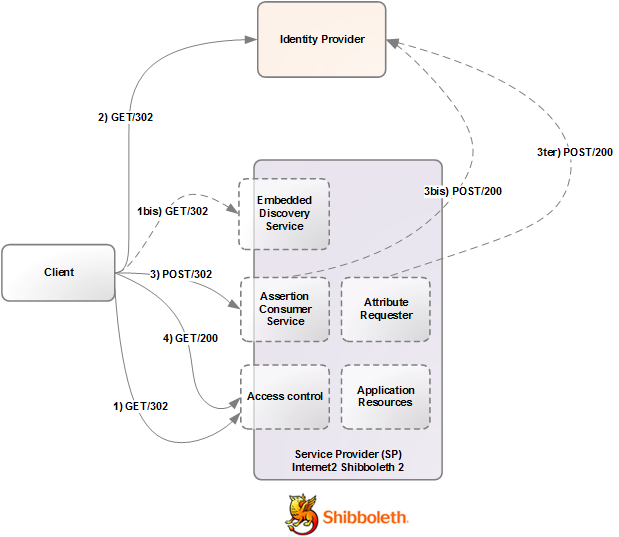
From an architecture perspective, a Shibboleth SP is made of the three following parts:

1. **Assertion Consumer Service** - this is the endpoint of the SP dedicated to the SSO part of the protocol. It behaves like a Web filter and redirects an unauthenticated user to his IdP so that he can authenticate.

**T**his may be done in conjunction with the optional Embedded Discovery Service (EDS) (formerly the WAYF service now deprecated with SAML 2.0) if it’s been implemented. The EDS allows the SP to run a discovery service within its own site. The discovery service conforms the [Identity Provider Discovery Service Protocol and Profile specification](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf)[[49]](#footnote-49) from the OASIS SAML TC. As such, the discovery service can look like any other page on the site and thus not redirecting the user to a totally different, third-party, discovery service site. The EDS is a set of JavaScript and CSS files, so installing it and using it is straight forward and does not require any additional software[[50]](#footnote-50).

Depending on the protocol profile, the assertion consumer service may either receive SAML assertions and optionally get additional attributes from the IdP, or receive a user artifact (a SAML artifact is a reference to a SAML assertion) and get user assertion and attributes from the IdP. It eventually redirects the client to the requested application resource. The collected attributes are transmitted to an access controller which decides whether or not the applicative resource may be accessed.

1. **Attribute Requester** - Based on a user artifact (used in some protocol profiles) this component is in charge of getting the user attributes from the IdP. This happens directly, without using the browser through redirections.
2. **Access control** - This component, as its name suggests it, is in charge of allowing or denying access to the applicative resource.



As of this writing, the current stable release of the Shibboleth SP is the [version 2.5.0 of the software](http://shibboleth.net/downloads/service-provider/latest/)[[51]](#footnote-51). The Shibboleth SP software is no longer covered in the rest of this document in so far as Windows Azure AD will play this role.

### Shibboleth Identity Provider (IdP)

On the other side, a Shibboleth IdP is composed of four elements:

1. **Authentication/SSO Service**: this service is in charge of authenticating a user who has been redirected to the IdP.

This is not an inner part of the IdP, but an IdP would be useless without a component that authenticates the user. This service must send to the IdP, more specifically the Authentication Authority a unique identifier for the user. This unique identifier will then be used to prepare and generate the user’s attributes. In practice, any mechanism (this includes the protocol and the users data source) available for web authentication may be used. This may be a Web SSO authentication system.

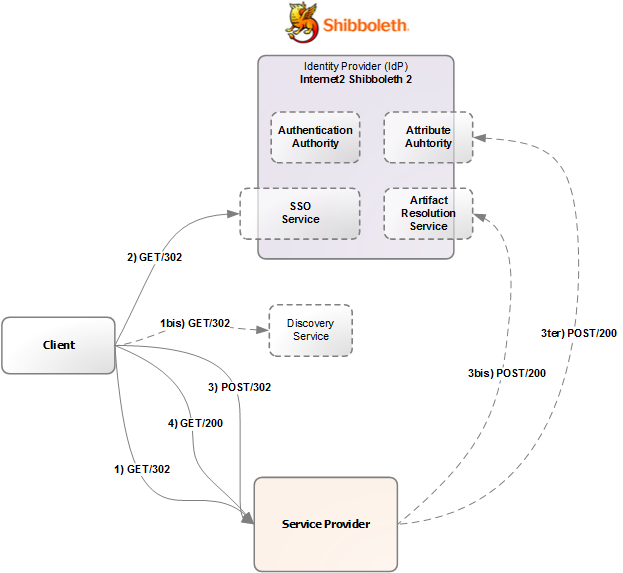


Indeed, integrating with the [CAS (Central Authentication Service)](http://www.jasig.org/cas)[[52]](#footnote-52) Web SSO service that was originally developed by Yale University[[53]](#footnote-53) is quite common; this is for instance the case among Universities in France. CAS is commonly used to only perform authentication and Web SSO.

**Note:**

*For instructions on how a Shibboleth IdP and CAS Web SSO can be brought together as a consistent solution, see the article* [*Shibboleth-CAS Integration*](https://wiki.jasig.org/display/CASUM/Shibboleth-CAS+Integration)*[[54]](#footnote-54). Further details about this are outside the scope of this document which concentrates on how Windows Azure AD/Office 365 can be a SP for a Shibboleth 2 identity federation.*

1. **Authentication Authority**: This authority is in charge of associating the name identifier artifact (an opaque identifier) with the preceding user unique identifier ;
2. **Attribute Authority**: This authority is the one which responds to SP invocations through a Web service; those invocations provide a user name identifier and expect the corresponding user attributes. The relationship between a user name identifier and the user attributes is maintained by the authentication authority. Different data sources (Active Directory, other LDAP directory, SQL databases, etc.) can be used as a user repository. The Shibboleth 2 implementation is able to handle several data sources at once.
3. **Artifact Resolution Service**: This service responds to a SP invocations through Web services, in relationship with the translation of a user artifact into an authentication assertion.



The Shibboleth 2 IdP technical implementation is a Java Web application based on the Servlet 2.4 specification. It requires as such a Java EE servlet container such as Apache Tomcat 6 (NOT 7). Apache Web server may also be used in front of Tomcat.

The current stable release of the Shibboleth IdP is the [version 2.3.8 of the software](http://shibboleth.net/downloads/identity-provider/latest/)[[55]](#footnote-55). There are no previous stable releases at this time.

## Interaction principles and associated profiles

As mentioned before, Shibboleth 2 provides backward compatibility with Shibboleth 1.3 and consequently supports profiles that come from SAML 1.1.

More interestingly in our context is the aforementioned support of [SAML 2.0](http://go.microsoft.com/fwlink/?LinkId=193996)[[56]](#footnote-56).

As described in section § 2.1.4 SAML 2.0 Profiles, SAML 2.0 defines an important number of profiles; the ones we are mostly interested in this paper are as follows:

* Web Browser SSO profile,
* Single Logout profile,
* Enhanced Client or Proxy (ECP) profile.

To shortly illustrate the interaction principles; let’s consider the Web Browser SSO profile.

The exchange begins with a request to the SP side. This modern approach referred as SP-initiated provides greater flexibility but rises the so-called Identity Provider Discovery problem in the SAML 2.0 jargon, as a reference to the eponym profile. This is also referred to as the Where Are You From (WAYF) and the Home Realm Discovery (HRD) issues. (The way IdP is selected,WAYF for example, is not defined.)

**Note:**

*Whilst the SP-initiated is a new addition and the intended approach, it should be mentioned that the former IdP-initiated approach as introduced by the previous version of SAML is still supported, but is no longer the preferred one.*

To illustrate the flexibility or simply to number of possible deployments resulting from the possible combinations within the profile, initial redirection from the SP to the IdP is not only possible through an HTTP redirect but also through an HTTP POST or a binding called HTTP artifact (it is based on HTTP redirects and SOAP message exchanges through HTTP). Consequently, a SP can choose from four bindings (HTTP Redirect, HTTP POST and two flavors of HTTP Artifact).

The authentication assertion is sent either through HTTP POST, either through the HTTP Artifact binding. The IdP has consequently three binding options (HTTP POST plus two forms of HTTP Artifact).

This conducts to a total of 12 possible deployments of the Web Browser SSO profile.

To take on example, the “SP-initiated SSO: Redirect/POST Bindings” describes an SP-initiated SSO exchange where the HTTP Redirect Binding is used to deliver the *<AuthnRequest>* message to the IdP and the HTTP POST Binding is used to return the *<Response>* message containing the assertion to the SP.

Section § 5.2 Understanding the passive/Web profile authentication flow later in this document more specifically covers the deployment of this profile in the specific context of the single sign-on with Windows Azure AD. See this section for additional details.

Likewise, section § 5.3 Understanding the ECP profile authentication flow specifically covers the deployment of the ECP profile, whose support is optional in the specific context of the single sign-on with Windows Azure AD.

(We do not cover the Single Logout profile even if the logout feature is part of the described configuration.)

## Federation metadata defined

With the Shibboleth 2 system, the different trust relationships between the members of a federation are implemented as (federation) metadata. They are the foundation of a federation trust as per the SAML 2.0 SAMLMeta document.

In order for an SP to be recognized by an IdP federation, it must be listed in the federation metadata. Similarly, in order for an IdP to be recognized by the SP inside the federation, it must be listed in the federation metadata. A Shibboleth SP may recognize only a subset of the IdP inside the federation, by defining a white list. Besides, for local needs, an IdP and an SP may trust each other, by exchanging their metadata files.

The Shibboleth 2 technical implementation automatically publishes its metadata at the following URL: *http(s)://<FQDN hostname>/idp/profile/Metadata/SAML*. This URL can be used for mutual trust relationships with an IdP or to register inside a federation.

Each provider’s metadata contains the provider’s unique identifier (entityID attribute) as a URN (Uniform Resource Name*,* Cf. [RFC 2141 URN Syntax](http://tools.ietf.org/html/rfc2141)[[57]](#footnote-57)), organizational and technical contact addresses, the name of the certificate this provider uses, the certificate authority URL and the attributes authority URL (for attribute requests) for an IdP, or the assertion consumer service URL (to receive attributes) for an SP.

Metadata also include certificate authorities (CA) which emit certificates that are trusted inside the federation.

**Note:**

*For more information, see the online help topic* [*NativeSPMetadataProvider*](https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPMetadataProvider)*[[58]](#footnote-58) on the Shibboleth Community wiki.*

Those metadata are usually expressed as an XML file. In order to ensure its integrity, this file is usually digitally signed. In a production federation, recording a new provider or updating information about an existing provider goes through a process that ensures the request is valid and legitimate.

In a federation, the XML file is managed centrally and it is used by all the IdP and SP inside the federation to mutually trust themselves. This aspect is covered later in this document.

# Federated authentication in Windows Azure AD/Office 365

The option to configure Shibboleth 2 is up to each individual company and knowing the expected behavior and experience that you will get is important. With the exception of Internet sites for anonymous access created with SharePoint Online, users must be authenticated when accessing the services in Office 365.

For that purpose, Windows Azure AD offers two types of identities:

1. **Microsoft Online Services cloud IDs (Cloud Identity)**: Users receive, for signing into Windows Azure AD, cloud credentials that are separate from other desktop or corporate on-premises credentials. The Cloud Identities are mastered in the service/cloud.

**Note:**

*With the optional directory synchronization,* *the user IDs mastered on-premises can be synchronized to the service/cloud in the form of Cloud Identities.*

1. **Federated IDs (Federated Identity)**: In companies with on-premises identities, the aforementioned single sign-on feature can be leveraged. Users can then sign into Windows Azure AD using their own corporate credentials to access services in Office 365. The user’s IDs are mastered on-premises and synchronized to the service in the form of Federated Identities.

Users can gain access to the services in Office 365 by authenticating to their Windows Azure AD user accounts, either through a prompt to provide valid credentials or through a single sign-on process if the CAS (Central Authentication Service) Web SSO service has been configured). Once authenticated, users’ identities refer to the user names associated with the Windows Azure AD accounts. Considering the above, we have three authentication types available:

1. Cloud Identities;
2. Cloud Identities + Directory Synchronization (DirSync Tool or Windows Azure AD connector for FIM 2010);
3. Federated Identities + Directory Synchronization (DirSync Tool or Windows Azure AD connector for FIM 2010).

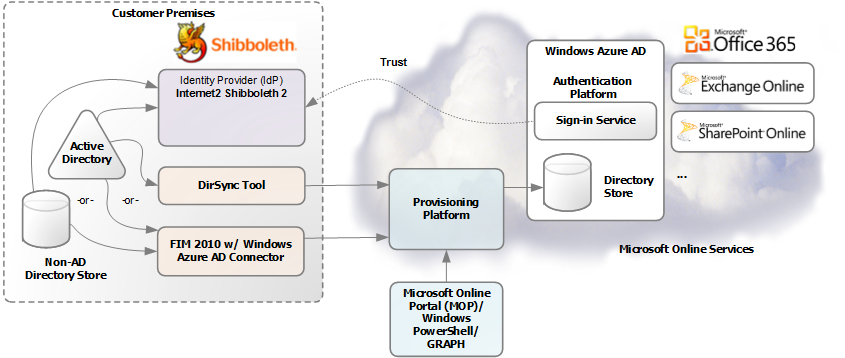
The above type of identity (cloud vs. federated) affects the user experience, the administrative requirements, the deployment considerations, and the capabilities using Windows Azure AD and the services in Office 365.

The following is the simplified breakdown of the experience:

* **User Experience with Cloud Identities**: users sign in with their cloud identity. Cloud Identities are authenticated using traditional challenge/response, where users type in their user name and password. Authentication happens in the cloud. Users are always prompted for credentials.

As mentioned above, users have two IDs, i.e. one to access on-premises services and one for Windows Azure AD, i.e. the cloud ID. Consequently, users are prompted for credentials even when logged into their on-premises environment when accessing the services in Office 365. This can actually be mitigated by selecting the save password option when you are prompted in many cases.

* **User Experience with Federated Identities**: users sign in with corporate ID for access to online and corporate services. In other words, they are authenticated using Shibboleth 2 when accessing the services in Office 365. Authentication happens on-premises against the organization’s identity environment. Users get true SSO if the CAS Web SSO service has been configured.
* **Administrator Experience with Cloud Identities**: organization’s administrators manage the password policy both in cloud and on-premises. The Cloud Identities password policy is stored in the cloud with the Windows Azure AD. Password reset has to be managed for on-premises and cloud IDs and hence the users have to change the password as per the policy for both.
* **Administrator Experience with Federated Identities**: Organization’s administrators manage the password policy on-premises only and hence do not need to separately worry about password reset for Federated Identities. The organization’s identity infrastructure stores and controls the password policy. Password reset occurs for on-premises IDs only.



The rest of this document discusses the single sign-on feature and the Federated Identities in this context. Consequently, for specific information on Windows Azure AD Cloud Identities such as user account creation, password policy, etc., please refer to the paper entitled [Office 365 Identity Service Description](http://www.microsoft.com/download/en/details.aspx?id=13602)[[59]](#footnote-59) as a starting point.

## Sign-in Experience for Federated Identities

The sign-in experience changes depending on the type of Windows Azure AD identity in use. The end-user sign-in experience varies depending on the client types.

The following table discusses the key combinations and helps explaining the resulting experiences. Please note that only a limited set of clients are supported in this single sign-on scenario.

|  |  |  |
| --- | --- | --- |
| **Application** | **Inside the corporate network** | **Outside the corporate network** |
| Microsoft Online Portal, SharePoint Online, Office Web Apps | Pop up offers click to sign in and prompted for credentials | |
| Outlook Web Apps | Prompted for credentials | |
| Office 2010/Office 2007 applications with SharePoint Online | Pop up offers click to sign in and prompted for credentials | |
| Outlook 2010/Outlook 2007, Exchange ActiveSync, POP, IMAP | Prompted for credentials on first connection (and at each password change) with checkbox to remember them. | |

Outlook users will be prompted to enter their corporate credentials on first use, at which time they can choose to save their password for future use. In this case, end-users will not be prompted again until they change their password, which depends on the organization’s password policies.

Integrating the Shibboleth 2 IdP with the CAS (Central Authentication Services) Web SSO service can further enhance the sign-in experience.

## Types of authentication for Federated Identities

This section discusses the types of user authentication that work with Windows Azure AD and the services in Office 365 for a Federated Identity.

The following table presents the several authentication mechanisms with a Federated identity.

| **Application** | **Authentication Mechanism** |
| --- | --- |
| Web browser | Web sign in, SAML 2.0 Web Browser SSO profile (Shibboleth 2) |
| Microsoft Office 2007/Office 2010 (Word, Excel, and PowerPoint) | Web sign in, SAML 2.0 Web Browser SSO profile (Shibboleth 2) |
| Outlook 2007/Outlook 2010, Exchange ActiveSync, POP/IMAP/SMTP client | Sign in, SAML 2.0 ECP profile (Shibboleth 2) |

### Authenticating from a Web Browser

Office 365 offers several services that you can access using a Web browser, including the Microsoft Online Portal (MOP), the preview of the Windows Azure Active Directory management portal, SharePoint Online, and Outlook Web App (OWA).

The single sign-on scenario with Shibboleth 2 supports the above Web-based services. This also includes Office 2007 or 2010 applications (Word, Excel and PowerPoint) to access SharePoint Online resources.

When you access these services, your browser is redirected to a sign-in page where you provide your sign-in credentials.

The sign-in experience is as follows for a Federated Identity:

1. The web browser is redirected to the Windows Azure AD sign-in service, where you type, for instance with an on-premises Active Directory environment, your corporate ID in the form a [User Principal Name (UPN)](http://msdn.microsoft.com/en-us/library/windows/desktop/ms680857(v=vs.85).aspx)[[60]](#footnote-60) formatted per IETF standard [RFC 822 Standard for ARPA Internet Text Messages](http://tools.ietf.org/html/rfc822)[[61]](#footnote-61), for example, [johndoe@idmgt.com](mailto:johndoe@contoso.com). The Windows Azure AD sign-in service determines that you are part of a federated domain and offers to redirect you to the on-premises Shibboleth 2 IdP service for authentication.
2. You then authenticate against the Shibboleth 2 IdP and Shibboleth 2 IdP generates a SAML 2.0 logon token, which the Web browser posts to the Windows Azure AD sign-in service. Using the logon token, the sign-in service generates an authentication token that the Web browser posts to the requested service in Office 365 and logs you in.

### Authenticating from Rich Client Applications

In this single sign-on scenario with Shibboleth 2, only e-mail rich clients that use basic authentication and a supported Exchange access method such as IMAP, POP, Exchange ActiveSync (EAS), MAPI, etc., are supported including:

* Microsoft Outlook 2007,
* Microsoft Outlook 2010,
* Thunderbird 8 and 9,
* The iPhone (various iOS versions),
* Windows Phone 7.x.

**All other clients are not supported in this single sign-on scenario with Shibboleth 2 IdP.  For example, the Lync 2010 desktop client is not supported to login into the service with the Shibboleth 2 IdP configured for single sign-on with Windows Azure AD.**

**The Enhanced Client Protocol (ECP) endpoint is required to be deployed for the Shibboleth 2 IdP.**

# Understanding the SSO configuration and related considerations

All the installation steps that have to be performed to setup the single sign-on feature with Shibboleth 2 along with the associated options are available and documented in the Microsoft TechNet articles [Single sign-on roadmap](http://technet.microsoft.com/en-us/library/hh967643)[[62]](#footnote-62) and [Use Shibboleth Identity Provider to implement single sign-on](http://technet.microsoft.com/en-us/library/jj205456)[[63]](#footnote-63).

## Preparing for the single sign-on

The Microsoft TechNet article [Prepare for single sign-on](http://technet.microsoft.com/en-us/library/jj151786)[[64]](#footnote-64) covers the operations that must be conducted in order to prepare the on-premise organization’s IT environment for single sign-on and a successful deployment of Federated Identities. Unsurprisingly, the preparation of the on-premises environment inevitably differs whether the organization uses Active Directory or non-AD directories.

This is something to take into account with Shibboleth 2, which can leverage different data sources (Active Directory, other LDAP directory, SQL databases, etc.).

### Shibboleth 2 in an on-premises AD environment

As described in the above Microsoft TechNet article, the organization’s on-premises Active Directory must indeed have certain settings configured in terms of both the structure and the use of the Active Directory domain name in order to work properly with single sign-on.

To prepare the Active Directory environment, we highly recommend running the [Microsoft Office 365 for enterprises Deployment Readiness Tool](http://g.microsoftonline.com/0BD00en-US/506)[[65]](#footnote-65) (*Office365DeploymentReadinessTool.exe*) that accompanies the [Microsoft Office 365 Deployment Guide](http://community.office365.com/en-us/f/183/p/1541/5095.aspx#5095)[[66]](#footnote-66).

This tool inspects the Active Directory environment and provides a report that includes information about whether or not you are ready to set up single sign-on. If not, it lists the changes you need to make to prepare for single sign-on.

A relevant assessment should cover the following topics:

1. **UPNs:** Federated Identities require corporate users to have a user principal name (UPN), though Active Directory does not. UPNs associate user’s identities in Windows Azure AD for enterprises with the identity in the cloud. Without this value, users may not be able to sign into Windows Azure AD with their corporate credentials.

UPNs that are used for Federated Identities can contain letters, numbers, periods, dashes, and underscores; no other types of characters are permitted. In addition, UPNs cannot end in a period before the at sign (@). In order to address UPN issues, there are a few options for modifying users in bulk. A few tools can be used for this operation such as [ADModify](http://admodify.codeplex.com/)[[67]](#footnote-67).

**Matching domains**: When Windows Azure AD domain names match the domain names for the local Active Directory, no special considerations regarding the name space are required.

1. **Sub domains:** Configure top-level domains first, and then configure sub-domains. When you register your top-level domain such as idmgt.fr, there is no need to register the subdomains such as legal.idmgt.fr or paris.idmgt.fr. Sub domains are automatically registered for you.
2. **Local Domains:** Local domains that are configured as .local (for example, idmgt.local) or .int (for example idmgt.int) cannot be used for federation because they cannot be accessed from the Internet (that is, they are not publicly routable or identifiable in DNS). You can register public domains with your registrar and then federate that domain with Windows Azure AD. Then add the new domain as a UPN domain suffix to your forest, and specify UPNs under the new domain. This will ensure that your federated users’ UPN domain suffix is under the domain that you federated with Windows Azure AD.
3. **Multiple forests:** “simple” multi-forests are supported for Federated Identities as previously categorized described (see section § 1.2 Non-objectives of this paper).

**“Complex” multi-forest scenarios can be achieved with Microsoft Premier Deployment (MPD) only.** **Please contact your local Microsoft Office 365 CXP Solution Architect if “complex” multi-forest support is a requirement for your organization.**

### Shibboleth 2 in an on-premises non-AD directories environment

Some organizations that have in-place directories other than Active Directory such as another LDAP directory and a SAML 2.0 based STS such as a Shibboleth 2 IdP desire to use Office 365 AND continue to use their existing on-premises infrastructure.

Such organizations typically need more preparation and require guidance that pertains to their own specific environment. The goal of this phase is to generate a clear process that shows how the on-premises identity infrastructure will integrate with Windows Azure AD provisioning, synchronization and single sign-on (identity federation).

In such a context, FIM 2010 and the Windows Azure AD Connector for FIM 2010 are required for the provisioning and synchronization parts as well the connectors to the identity sources in place. See the article [Management Agents in FIM 2010](http://technet.microsoft.com/en-us/library/ff608275(WS.10).aspx)[[68]](#footnote-68) for a list of FIM 2010 supported identity sources. In addition to this list, the OpenLDAP Extensible Management Agent (XMA) connector for FIM 2010 is available on [SourceForge](http://sourceforge.net/projects/openldap-xma/)[[69]](#footnote-69) as an open source project for OpenLDAP directories.

In such a situation, the use of a dedicated FIM 2010 instance is recommended, to reduce complexity.

**As for multi-forest support with AD, availability of this solution is currently subject to same pre-requisites. Please contact your local Microsoft Office 365 CXP Solution Architect if Non-AD directories are a requirement for your organization.**

## Planning and deploying a Shibboleth 2 identity provider

**As described in section § 2 A brief overview of Shibboleth 2, Shibboleth 2 is a third-party product and therefore Microsoft does not provide support for the deployment, configuration, troubleshooting, best practices, etc. issues and questions regarding Shibboleth 2.**

This said, in order to simplify the implementation stage in a testing environment, this section describes the main steps relating to the installation and the configuration of a Shibboleth 2 IdP. As such, it contain instructions for local administrators of a Microsoft cloud service such as Office 365 who want to provide their Windows Azure AD users with single sign-on (SSO) experience by using a corporate Shibboleth 2 IdP as their preferred Security Token Service (STS).

**The information provided is based on our experience in setting up a test Shibboleth 2 implementation, and is not intended to be an exhaustive guide to installing and configuring Shibboleth 2.**

It is based on the definition of one on-premises standalone machine running under Windows Server 2008 R2 with the Shibboleth Identity Provider software.

**Note:**

*Please note that running the Shibboleth environment on a Windows system is not a requirement. There are many guides that describe the installation for a Linux system.*

**Note:**

*Microsoft has published, thanks to author Dave Martinez, the* [*AD FS 2.0 Step-by-Step Guide: Federation with Shibboleth 2 and the InCommon Federation*](http://go.microsoft.com/fwlink/?LinkId=204784)*[[70]](#footnote-70) in a series of step-by-step guides[[71]](#footnote-71) on configuring AD FS 2.0 to interoperate with partner products. This guide describes how to configure Shibboleth 2 to federate using the SAML 2.0 protocol with the SAML2.0 HTTP POST binding. The current document leverages some of the information contain in this guide.*

To easily replicate the configuration illustrated hereafter in a testing environment:

* You must follow the steps described in the two following chapters in the order in which these steps appear in order to have, at the end, hopefully a fully operational testing environment.

These steps assume the prior existence of the on-premises domain *shib*.*idmgt.archims.fr*.

Generally speaking, they also assume that the considered Windows Server 2008 R2 environments have all the prerequisites required to the installation of Shibboleth 2 IdP software and therefore do not necessary describe the installation of these prerequisites.

* You can create your own virtual machine for IDP0 (*idp0*.*shib*.*idmgt.archims.fr*) that you will configure later to perform the tasks relating to the steps outlined in the rest of this document.

In order to do that, if you do not already have clean-installed Windows Server 2008 R2 virtual hard drive image, you can download and use the base evaluation .VHD files to build the base VMs for this lab. The files are available on the Microsoft Download Center at [Windows Server 2008 R2 Evaluation Virtual Hard Drive Images for Hyper-V (180 Days)](http://go.microsoft.com/fwlink/?LinkId=179736)[[72]](#footnote-72).

The steps described in the following sections assume the availability of the IDP0 (*idp0.shib.idmgt.archims.fr*) Windows Server 2008 R2-based (virtual) machine on which the Shibboleth IdP software will be installed:

* Host operating system: Windows Server 2008 R2 Enterprise Edition.
* Additional feature: .NET Framework 3.51.

Unless noted otherwise, all the instructions below are executed on this machine.

For information on the prerequisites as well as on the installation and the configuration of a Shibboleth IdP, we invite you to visit complementarily the documentation made available by the [Shibboleth Community](https://wiki.shibboleth.net/confluence/display/SHIB2/Installation)[[73]](#footnote-73). (Both Windows and Linux environments are covered.) Please also note that you must [configure](https://wiki.shibboleth.net/confluence/display/SHIB2/Configuration)[[74]](#footnote-74) the software after installation.

### Performing the pre-configuration tasks

Perform the followings prerequisite configuration steps to prepare your testing on-premises environment for testing.

**Note:**

*All of the actions in this section were performed while logged into Windows with administrative privileges.*

#### Ensuring IP Connectivity

Make sure that the Shibboleth IdP (*idp0.shib.idmgt.archims.fr*) computer has Internet connectivity.

#### Configuring Name Resolution

The Shibboleth 2 IdP (*idp0.shib.idmgt.archims.fr*) computer must be accessible by name from the Internet, and the domain name against which it authenticates must be a valid Internet domain name.

The Shibboleth 2 IdP computer can be placed in a demilitarized zone (DMZ) and the appropriate ports must be opened on the internal and external firewalls.

Only ports 80, 443 (Shibboleth Client port), and 8443 (Shibboleth Back Channel port) needs to be opened for successful Windows azure AD authentication.

When installing the Shibboleth 2 IdP with Windows .msi software package, the local Windows Firewall is automatically configured with the appropriate inbound rules. This is one of the benefits of using this installer.

#### Verifying Clock Synchronization

Federation events typically have a short Time to Live (TTL). To avoid errors based on time-outs, ensure that both computers have their clocks synchronized.

**Note:**

*For information about how to synchronize a Windows Server 2008 R2 domain controller to an Internet time server, see Knowledge Base article* [*816042 How to configure an authoritative time server in Windows Server*](http://support.microsoft.com/kb/816042)*[[75]](#footnote-75).*

### Installing the Java Runtime Environment

As previously mentioned, the Shibboleth 2 IdP technical implementation is a Java Web application. It consequently requires a Java Enterprise Edition (Java EE) servlet container such as Apache Tomcat 6 (NOT 7), which is the most commonly used supported container. (Other supported containers are JBoss-Tomcat and [Jetty](http://jetty.mortbay.org/jetty)[[76]](#footnote-76), a free Java Web server.) In turn, the Java EE servlet container requires a Java Runtime Environment (JRE) to run.

The Shibboleth 2 IdP Windows .msi installation package installs a "captive" Tomcat for you, so you only need to have Java installed before you start. This installer results in a standard Tomcat-only Shibboleth 2 installation, and takes a lot of the effort out of the configuration.

**Note:**

*For other platforms, you need to start with Java and the Java EE servlet container being installed. Some organizations use the Apache httpd web server as a proxy front end, although this is no longer necessary, nor recommended by the Shibboleth 2 developers.* *In general, the most recent software versions should be used, but ensure that you check the Shibboleth 2 IdP documentation for any caveats.*

The Shibboleth 2 IdP Windows .msi software package requires that you start out with a recent 32-bit version (7 or later) of the JRE only. All other required software is bundled with the installer.

This document uses [Oracle Java SE Runtime Environment (JRE) 7u7](https://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_Developer-Site/en_US/-/USD/ViewProductDetail-Start?ProductRef=jre-6u19-oth-JPR@CDS-CDS_Developer)[[77]](#footnote-77) for Windows **32-bit**.

procedure_ddTo install the JRE environment, proceed with the following steps:

1. Navigate to Java Web site at <http://java.com/en/download/ie_manual.jsp?locale=en> and click **Agree and Start Free Download** to install the JRE environment.



1. Click **Run**. The **Java Setup – Welcome** dialog opens up.



1. Click **Install >**. The installer is downloaded. The **Java Setup** dialog opens up and automatically proceeds with the installation.



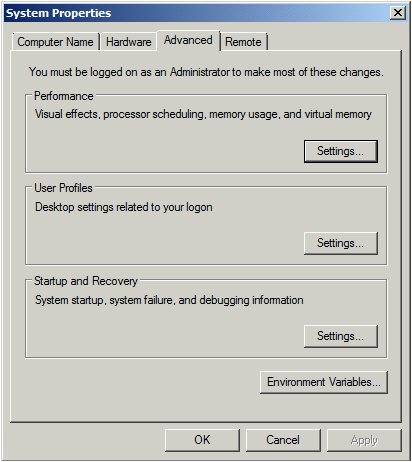
1. Once JRE installed, click **Close**.

The JAVA\_HOME system environment variable must be set to the install directory for this JRE (for example, *C:\Program Files (x86)\Java\jre7*) before running the Shibboleth 2 IdP software installer.

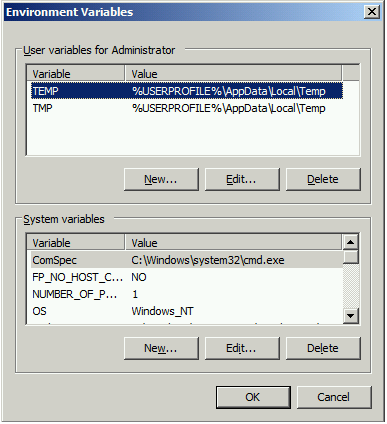
JAVA\_HOME=C:\Program Files (x86)\Java\jre7

procedure_ddTo set the JAVA\_HOME variable, proceed with the following steps:

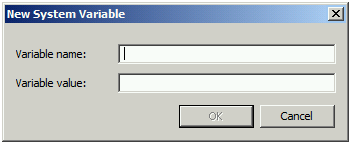
1. Click **Start**, right-click **Computer** and select **Properties**.
2. Select **Advanced System Settings**. The **System Properties** dialog opens up.



1. From the **Advanced** tab, click **Environment Variables…** The **Environment Variables** dialog opens up.



1. Under the **System variables** section, click **New…** The **New System Variable** dialog opens up.



1. Enter “*JAVA\_HOME*” in **Variable name** and “*C:\Program Files (x86)\Java\jre7*” in **Variable value**, and then click **OK** to close the **New System Variable** dialog.
2. Click **OK** to close the **Environment Variables** dialog.
3. Click **OK** to close the **System Properties** dialog.

### Requesting a SSL certificate

Federation relies heavily on public key infrastructure (PKI), including Secure Sockets Layer (SSL) encryption, for trustworthy transactions.

The Shibboleth 2 IdP installation process (covered later in this document, see section § 4.2.5 Installing the Shibboleth 2 IdP software) generates a long-life self-signed certificate used by default for the SSL connection to the Shibboleth 2 IdP (and consequently by the Tomcat server underneath) and for SAML token signing. The common name (CN) of the certificate matches the DNS hostname of the Shibboleth 2 IdP server.

Even if this has become the recommended approach for Shibboleth, service providers in some federation may not accept self-signed certificates. This also has implications regarding the trusted root certificates of the browser.

For our configuration, we illustrate as a consequence the situation where you request a certificate from a public Certificate Authority (CA) or from your own PKI.

In such case, please ensure that the CN of the certificate in the request match the DNS hostname of the intended Shibboleth 2 IdP server. A key length of at least 2048 bits is recommended for the certificate, 1024 bits being the minimum for the length of the RSA key.

Request a SSL certificate for *idp0.shib.idmgt.archims.fr*.from a public Certificate Authority (CA) or issue your own with your own PKI.

Let’s suppose you get back from the public CA a Personal Information Exchange (.pfx) file, for example *idp0.shib.idmgt.archims.fr.pfx* in our configuration.

A .pfx file is an archive file format commonly used to directly store X.509 certificates and private keys, which conforms with the PKCS#12 standard. PKCS #12 is one of the families of standards called Public-Key Cryptography Standards (PKCS) published by RSA Laboratories. This extension could also be .pkcs12 or .p12.

Our *idp0.shib.idmgt.archims.fr.pfx* file combines our SSL certificate's public key file and the trust chain with their associated private key up to the Baltimore CyberTrust Root certificate.

In order to convert this file to a Java key store (.jks) usable by a Java-based server such as Tomcat in our context, we need to use some Java utilities such as the PKCS12import utility that is distributed as part of Jetty.

**Note:**

*For information, see the blog posts* [*Convert a PFX to JKS using Windows*](http://www.agrypnia.com/blog/2009/07/28/convert-a-pfx-to-jks-using-windows.html)*[[78]](#footnote-78) and* [*Private Key PFX to/from JKS Conversion Using OpenSSL and Jetty*](http://andrius.miasnikovas.lt/2010/10/private-key-pfx-jks-conversion-using-openssl-and-jetty/)*[[79]](#footnote-79).*

Description: procedure_ddTo create the .jks file from the .pfx file, proceed with the following step:

1. Download the *Jetty-6.1.16.zip* file from <http://docs.codehaus.org/display/JETTY/Downloading+Jetty>.
2. Simply extract the Jetty files to a folder, for example *C:\Jetty-6.1.16*. We will not actually be installing Jetty, just using the PKCS12import utility included with it.
3. Open a command prompt use and type the following command to test for existence of the Java class:

C:\> “%JAVA\_HOME%\bin\java” -classpath jetty-6.1.26/lib/jetty-6.1.26.jar org.mortbay.jetty.security.PKCS12Import

You should get a return message that looks like this:

usage: java PKCS12Import {pkcs12file} [newjksfile]

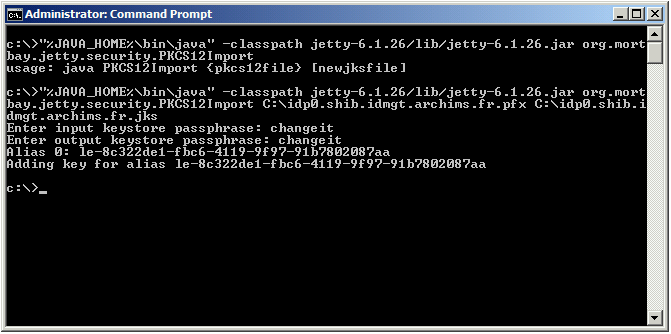
1. From the command prompt, type the following command to create the .jks file:

C:\> “%JAVA\_HOME%\bin\java” -classpath lib/jetty-6.1.26.jar org.mortbay.jetty.security.PKCS12Import C:\idp0.shib.idmgt.archims.fr.pfx C:\idp0.shib.idmgt.archims.fr.jks

Enter the password of the input .pfx file when prompted, for example “*changeit*” in our case.

Enter the password of the output .jks file when prompted, for example “*changeit*” in our case.

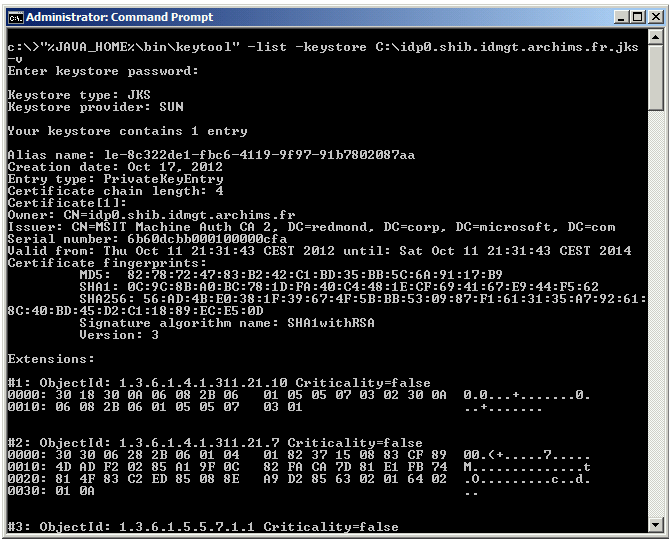
Here is what you should see after running the command:



1. Run this command to verify if the .jks file is valid:

C:\> “%JAVA\_HOME%\bin\keytool” -list -keystore C:\idp0.shib.idmgt.archims.fr.jks -V

Enter the password of the .jks file when prompted, for example “*changeit*” in our case. You should see something similar to the following:



To replace the generated self-signed certificate issued during the installation of the Shibboleth 2 IdP, we also need to generate:

1. A .crt file that contains our SSL certificate. (The extension can also be .cert, .cer or .pem. With the exception of the last one, all these extensions are recognized by Windows Explorer as a X.509 certificate.)
2. And a .key file for the private key of the SSL certificate.

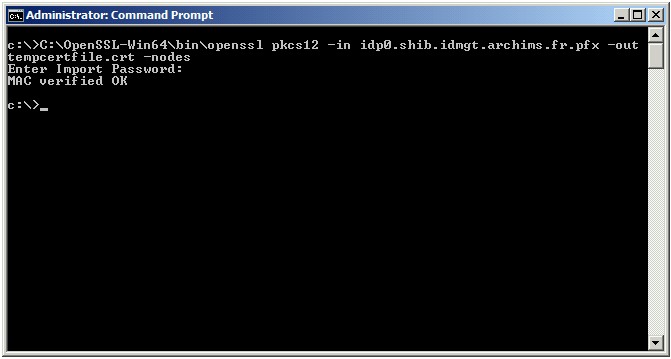
We also need to generate a .crt file for the root certificate to which our SSL certificate chains to. All of this can be achieved via the OpenSSL library.

Description: procedure_ddTo create the .crt and the .key files from the .pfx file, proceed with the following step:

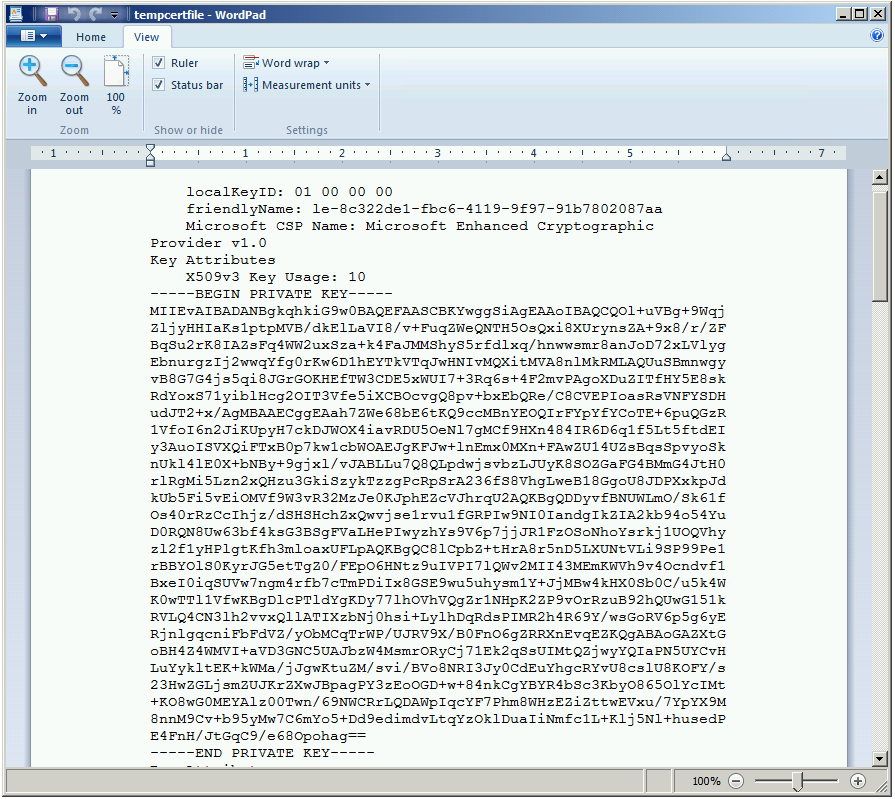
1. Download the OpenSSL installation project (*Win64OpenSSL-1\_0\_1c.exe*) from <http://slproweb.com/products/Win32OpenSSL.html> and execute it. By default, the OpenSSL library package is installed under *C:\OpenSSL-Win64*.
2. From a command, run the following OpenSSL command to extract the SSL certificate and its private key from the .pfx file:

C:\> C:\OpenSSL-Win64\bin\openssl pkcs12 -in idp0.shib.idmgt.archims.fr.pfx -out tempcertfile.crt –nodes

Enter the password of the input .pfx file when prompted, for example “*changeit*” in our case.



1. You should now have a file called *tempcertfile.crt*. Open this file with a text editor (such as WordPad) and you will see the private key listed first, followed by the certificate files:



-----BEGIN RSA PRIVATE KEY-----   
(Block of Encrypted Text)   
-----END RSA PRIVATE KEY-----

Cut and paste the first private key, including the BEGIN and END tags to a new text file and save it as *idp0.shib.idmgt.archims.fr.key*.

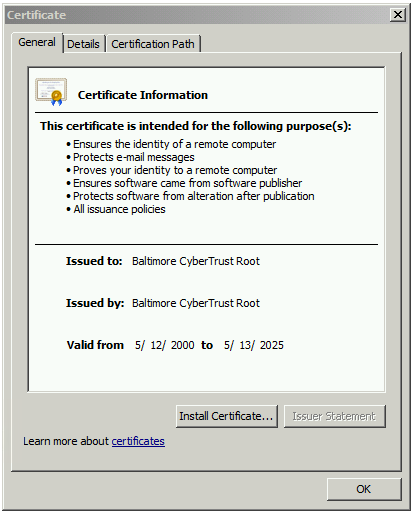
1. The certificates remaining in the *tempcertfile.crt* file will be in the following order, SSL certificate, CA root certificate, intermediate certificate; depending on your .pfx export, there could be between 2 and 4 separate certificates inside this one file.

As long as you exported the certificates correctly, whatever you have is what you are supposed to have. Make sure the private key was removed (not just copied and pasted), and then go ahead and save this file as *idp0.shib.idmgt.archims.fr.crt*.

1. From the *idp0.shib.idmgt.archims.fr.crt* file, copy the second set of -----BEGIN CERTIFICATE----- & -----END CERTIFICATE----- tags, and all the coded text in between into a separate file, and then save it as *rootca.crt*.

-----BEGIN CERTIFICATE-----   
(Block of Encrypted Text)   
-----END CERTIFICATE-----

1. Double click the *rootca.crt*. A Certificate dialog should open up and display the root certificate.



The "cacerts" key store file of the JRE stores all the trusted root certificates. In our configuration, it already ships with the above Baltimore CyberTrust Root certificate, so you probably won't need to import a Baltimore CyberTrust Root certificate as a trusted certificate in your key store. But if you request a signed certificate from a different CA, and a root certificate authenticating that CA's public key hasn't been added to "cacerts", you will need to import that root certificate from the CA as a "trusted certificate".

Description: procedure_ddIf needed, to import the root certificate of the issued SSL certificate as a trusted certificate in your key store in JRE, proceed with the following steps:

1. Execute the following from an elevated command prompt:

C:\> “%JAVA\_HOME%\bin\keytool.exe” –import –keystore “%JAVA\_HOME%\lib\security\cacerts” –file C:\rootca.crt

Enter the password as “*changeit*” (this is the key store default) and confirm with “*yes*”.

### Installing and configuring Active Directory Lightweight Directory Services

When running on a Windows platform, the Shibboleth 2 IdP software installer (pre-) supposes that the server will communicate with Active Directory to perform two primary functions:

1. Authenticate the users from Active Directory,
2. Query Active Directory for information about users and presents it to back to services providers such as Windows Azure AD in our context.

To allow the operation of these functions, the documentation of the software installer states that the server should be a member server within an Active Directory forest/domain. In this case, the Shibboleth 2 IdP software installation package is indeed able to detect that the computer on which it runs is member of an AD domain and the Control Information page of the installation package wizard is automatically populated with information about the host Active Directory.

Considering that Shibboleth 2 can rely on Active Directory infrastructure or non-AD directories, and that the Microsoft TechNet documentation already describes the configuration aspects of the Shibboleth 2 IdP with Active Directory, we do not want to leverage an Active Directory (AD) forest environment with Active Directory Domain Services (AD DS). We rather prefer opting for an approach that can directly be transposed to other LDAP repository like the [OpenLDAP software](http://www.openldap.org/)[[80]](#footnote-80) in order to also illustrate the situation where the organization uses non-AD directories.

Consequently, in order to have a LDAP repository other than Active Directory that will provide attributes and user accounts to the Shibboleth 2 IdP software, we’re going to install and configure Active Directory Lightweight Directory Services (AD LDS). (This is also the reason why the IDP0 virtual machine is configured as a standalone server.)

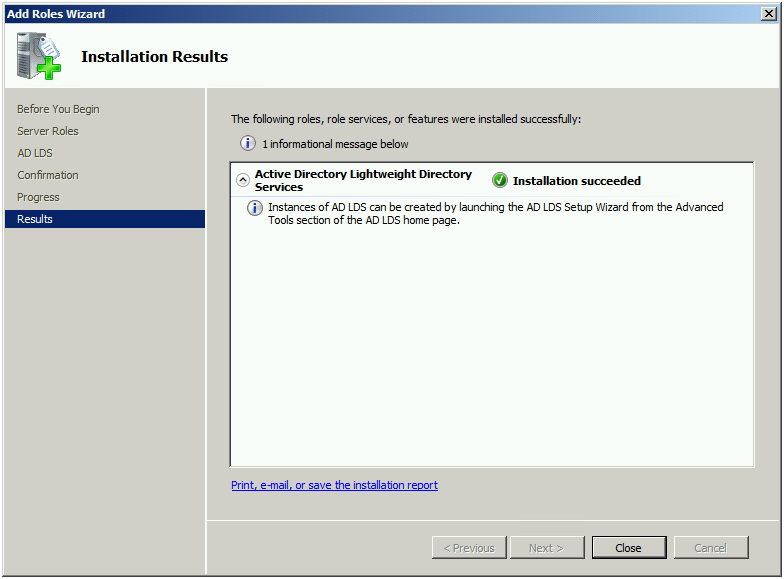
**Note:**

*For information about AD LDS, see the article* [*Active Directory Lightweight Directory Services*](http://technet.microsoft.com/en-us/library/cc731868(WS.10).aspx)*[[81]](#footnote-81) available on the Microsoft TechNet Web site.*

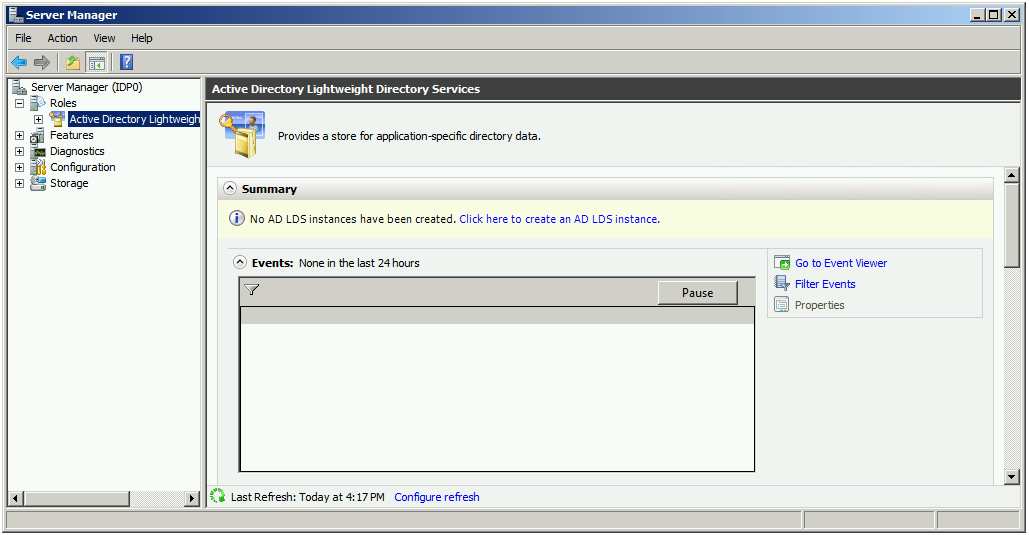
#### Installing the AD LDS role and configure the instance

Description : procedure_ddTo both install an AD LDS role and configure the instance onto the *idp0.shib.idmgt.archims.fr* Windows Server 2008 R2 machine and declare it for logon for the Shibboleth IdP, proceed with the following steps:

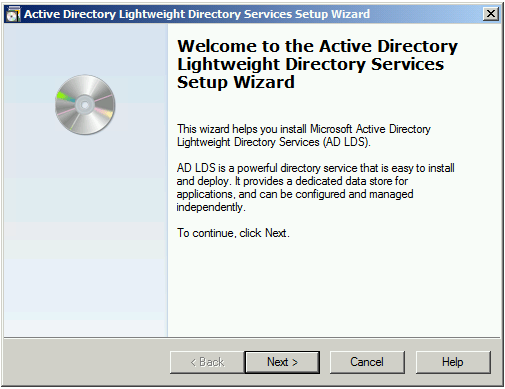
1. Add the AD LDS role to the Windows instance as explained in the [Step 1: Install the AD LDS Server Role page](http://technet.microsoft.com/en-us/library/cc754486(v=ws.10).aspx)[[82]](#footnote-82). This will end with the following result in the **Add Roles wizard**:



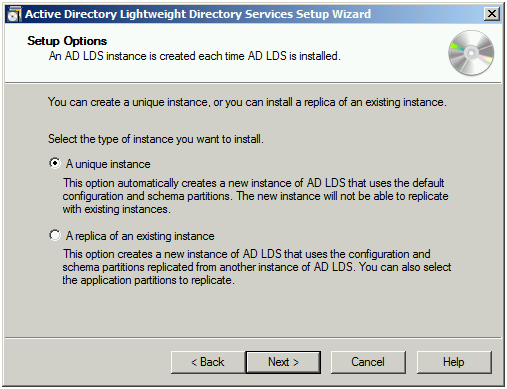
1. From the **Server Manager** console, select the **Active Directory Lightweight Directory Services** node in the console tree under the **Roles** node.



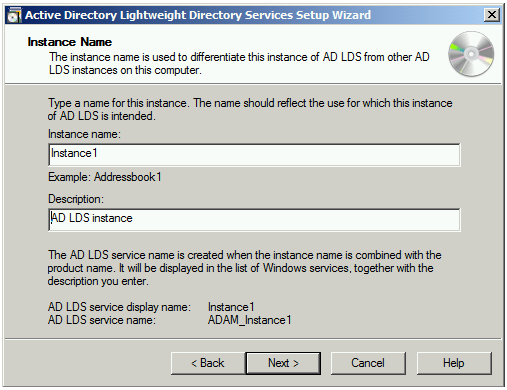
In the left pane, click **Click here to create an AD LDS instance**. The **Active Directory Lighweight Directory Services Setup** **Wizard** dialog opens up.



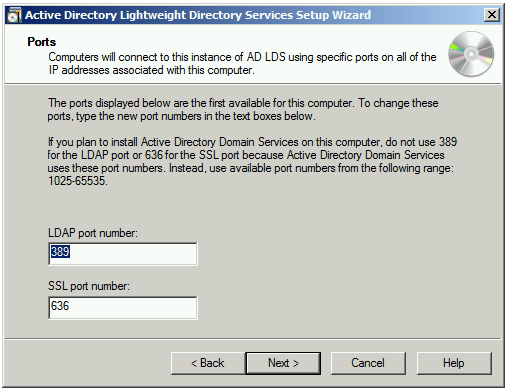
1. On the **Welcome** page, click **Next**.



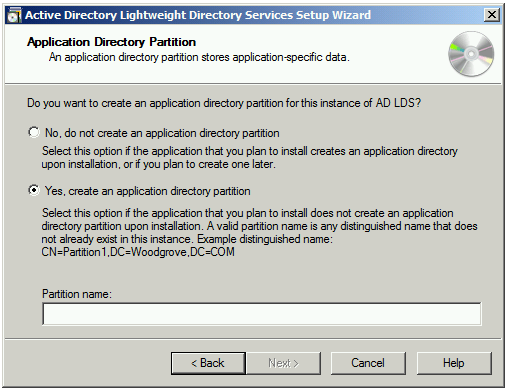
1. On the **Setup Options** page, select **A unique instance**, and then click **Next**.



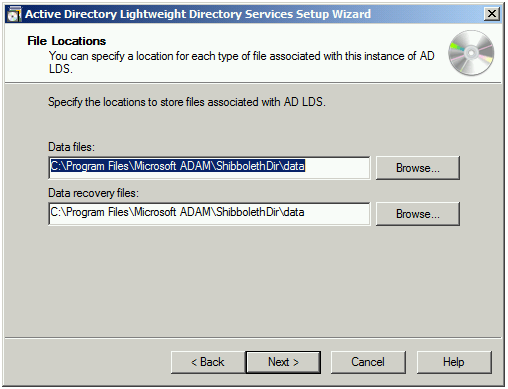
1. On the **Instance Name** page, type for example “*ShibbolethDir”* in **Instance name** and “*Shibboleth Directory AD LDS instance”* in **Description**, and then click **Next**.



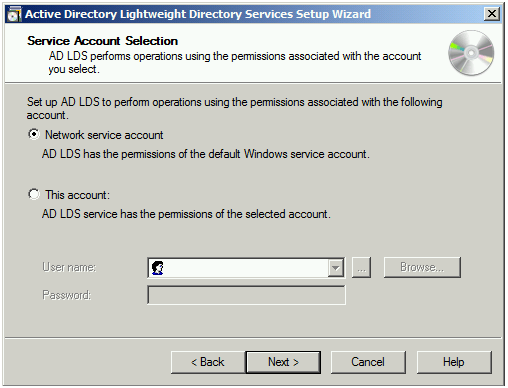
1. On the **Ports** page, keep the default port number for both LDAP and SSL, and then click **Next**.



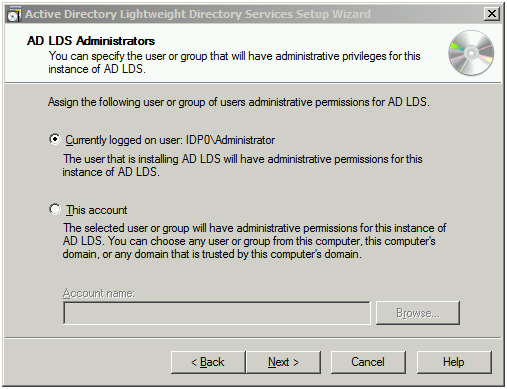
1. On the **Application Directory Partition** page, select **Yes, create an application directory partition** for the Shibboleth IdP, then type for example “*dc=SHIB,dc=IDMGT,dc=ARCHIMS,dc=FR”* in **Partition name**, and finally click **Next**.



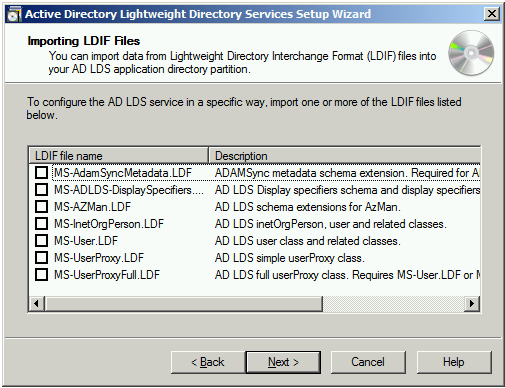
1. On the **File Locations** page, keep the default path for both the data files and the data recovery files, and then click **Next**.



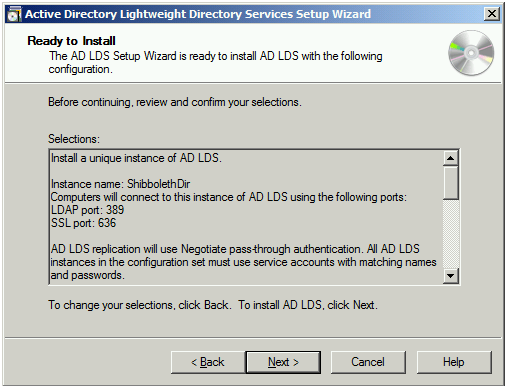
1. On the **Service Account Selection** page, keep the default network service account, and then click the **Next** button.



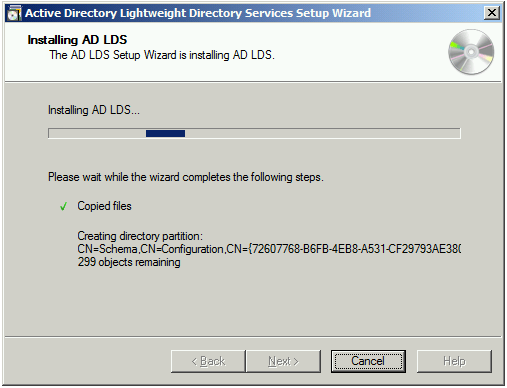
1. On the **AD LDS Administrators** page, keep the default currently logged on user as the instance administrator, and then click **Next**.



1. On the **Importing LDIF Files** page, make sure that the LDIF files *MS-InetOrgPerson.LDF* and *MS-User.LDF* are checked, and then click **Next**.



1. On the **Ready to Install** page, review the selections made so far, and then click **Next** to start the installation.



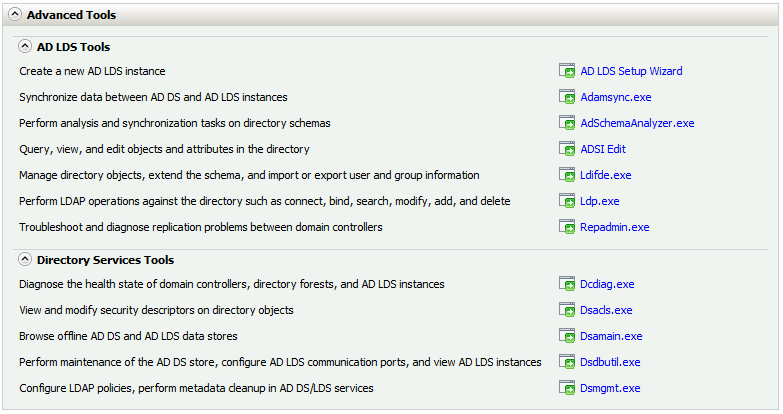
1. Once completed without any error, the completingpage appears. Click **Finish** to complete the instance installation.



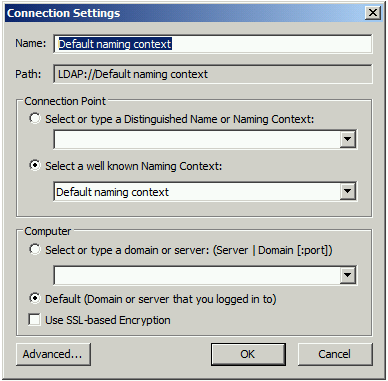
#### Connecting to the AD LDS instance with ADSI Edit

Description : procedure_ddTo connect to the AD LDS instance with ADSI Edit, proceed with the following steps:

1. From the **Server Manager** console, select the **Active Directory Lightweight Directory Services** node in the console tree under the **Roles** node.

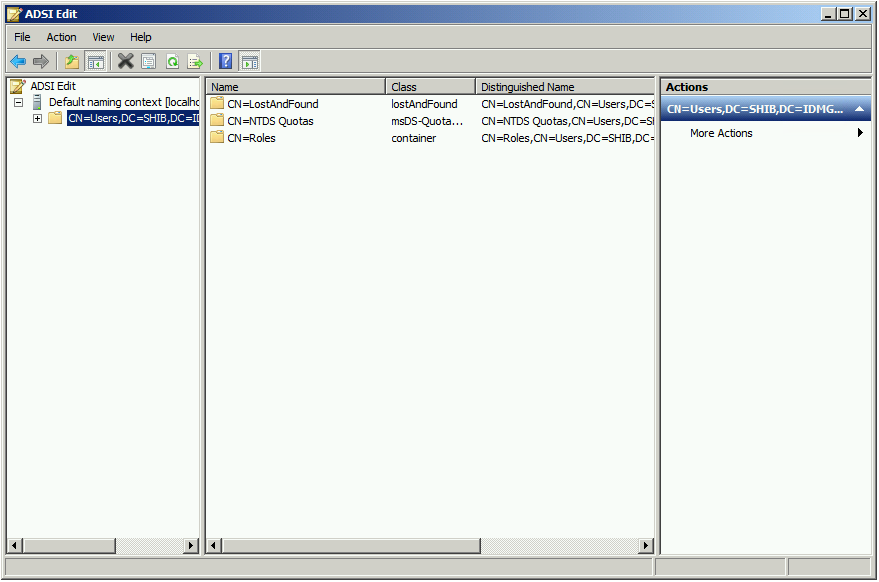


1. In the right pane, click the **ADSI Edit** link. The **ADSI Edit** application opens up.
2. Right-click the **ADSI Edit** node and select the **Connect to** item. The **Connection Settings** dialog opens up.



1. In the **Connection Point** frame, select **Select or type a Distinguished Name or Naming Context** and type the partition name specified earlier during the AD LDS instance installation, for example “*CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR”* in our case.

In the **Computer** frame, select **Select or type a domain or server: (Server | Domain [:port])**, then type “*localhost:389”*, and finally click **OK** to confirm the settings and close the dialog.



#### Creating a Shibboleth Service account

As stated above, a Shibboleth 2 IdP queries the configured identity source for information about users. To query an AD LDS instance in our configuration, the Shibboleth 2 IdP will need a set of credentials, i.e. a username and a password.

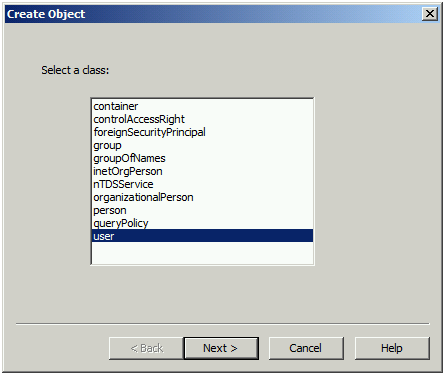
A dedicated user account should be used for the following reasons:

* It simplifies both the initial setup and on-going troubleshooting;
* It facilitates targeted security analysis, i.e. investigations can be concentrated on the actions of one particular account.

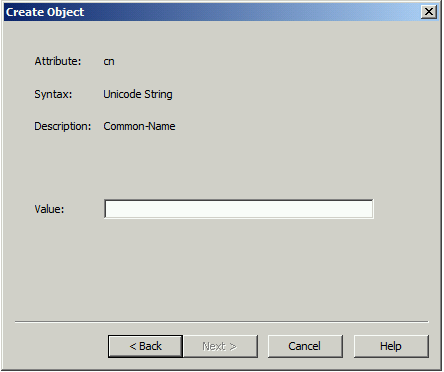
It is now time to create a user inside the LDAP directory so that the Shibboleth 2 IdP can access AD LDS with its credential.

Description : procedure_ddTo create a Shibboleth Service user in AD LDS, proceed with the following steps:

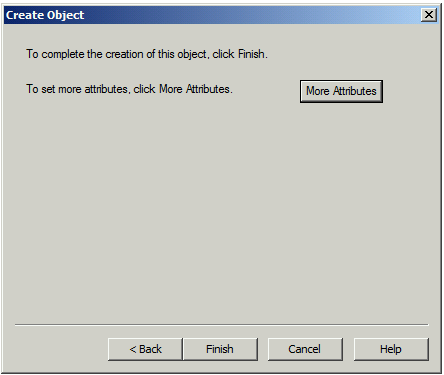
1. Connect to the AD LDS instance as per section § 4.2.4.2 Connecting to the AD LDS instance with ADSI Edit.
2. In the **ADSI Edit** console, right-click the “*CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR”* node, then click **New**, and then click **Object**. The **Create Object** dialog opens up.



1. In the **Create Object** dialog, select **User** and then click **Next**.



1. In the **Create Object** dialog, type for example “*ShibSvc”* for the value and then click **Next**.

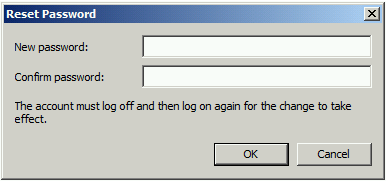


1. In the **Create Object** dialog, click **Finish** to create the user.

We then have to set the “*CN=ShibSvc”* user’s password.

Description : Description : procedure_ddTo set the user’s password, proceed with the following steps:

1. Right-click the newly created “*CN=ShibSvc”* user object and click **Reset Password**. The **Reset Password** dialog opens up.

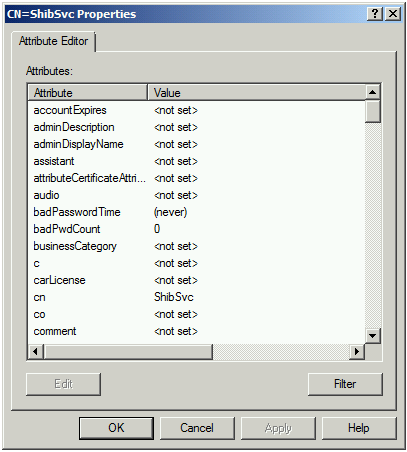


1. Specify the new password, for example “*Password1*”, in **New password** and **Confirm password** and then click **OK** to confirm and close the dialog.

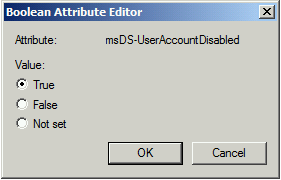
For the “*CN=ShibSvc*” user, the *msDS-UserAccountDisabled* attribute must be changed to false, and the *msDS-UserDontExpirePassword* attribute to true.

Description : Description : procedure_ddTo set these attributes, proceed with the following steps:

1. Right-click the “*CN=ShibSvc”* user object and click **Properties**. The properties dialog opens up.



1. Select the *msDS-UserAccountDisabled* attribute in the attributes list and then click **Edit**. The **Boolean Attribute Editor** dialog opens up.

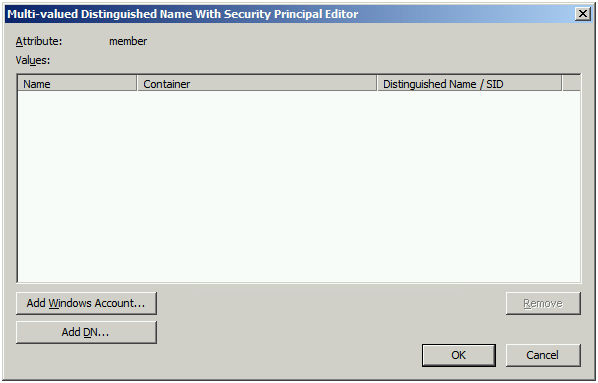


1. Select **False** and then click **OK** to close the editor box.
2. Similarly, set the *msDS-UserDontExpirePassword* attribute of the same account to true.
3. Click **OK** to close the properties dialog.

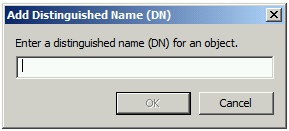
Finally the “*CN=ShibSvc”* user must belong to (at least) the “*CN=Readers”* role. (It can be a member of the “*CN=Administrators”* role instead.)

Description : Description : procedure_ddTo place the Shibboleth service user in the Readers role, proceed as follow:

1. Click the “*CN=Roles”* container in the left pane, then right-click the “*CN=Readers*” object in the right pane, and then click **Properties**. The properties dialog opens up.
2. Select **member** in the list and then click **Edit**. The **Multi-valued Distinguished Name With Security Principal Editor** dialog opens up.



1. Click **Add DN…**. The **Select Users or Groups** dialog opens up.



1. Type “*CN=ShibSvc,CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR*” in **Enter a distinguished name (DN) for an object.** , and then click **OK**.
2. Click **OK** to close the editor dialog.
3. Click **OK** to close the properties dialog.

#### Creating and configure a test user for the Shibboleth 2 IdP

In order to interact with Windows Azure AD in the other realm, a test user has to be created.

The required steps are similar as the one outlined in the previous section:

* Create a new object of type user, named “*user1*” instead of “*ShibSvc*” (“*CN=user1,CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR”*);
* Set its password to “*Password1*” for example;
* Set its *msDS-UserAccountDisabled* attribute to false.
* You may also want to set its *msDS-UserDontExpirePassword* attribute to true.

**Note:**

*This user doesn’t have to be added in any role.*

The following additional attributes are set (they will eventually be transformed as claims):

|  |  |
| --- | --- |
| **Attribute name** | **Value** |
| mail | *user1@shib.idmgt.archims.fr* |
| uid | *81372* |

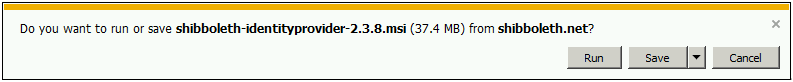
### Installing the Shibboleth 2 IdP software

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.

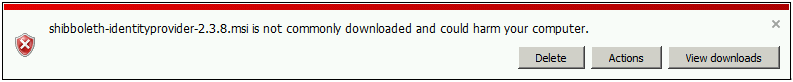
This enables to save a considerable amount of time, effort, and work at configuration time, so long as you enter the required information accurately at installation time. (Apache Tomcat 6 is included with the Windows installer, and Apache httpd is not required.)

procedure_ddTo install the Shibboleth IdP software package onto the IDP0 (*idp0.shib.idmgt.archims.fr*) machine, proceed with the following steps:

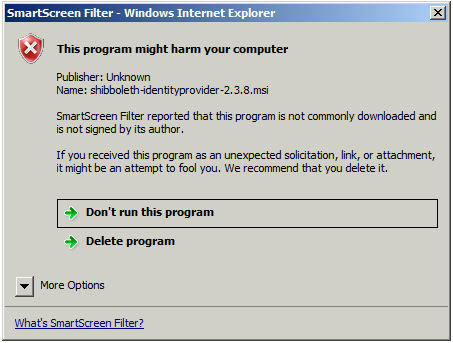
1. Visit the [Shibboleth IdP download site](http://go.microsoft.com/fwlink/?LinkId=204016)[[83]](#footnote-83) and download the latest IdP (Identity Provider) software package. This guide uses the following Windows .msi installation package: *shibboleth-identityprovider-2.3.8.msi*.



1. Click **Run** to execute *shibboleth-identityprovider-2.3.8.msi*.



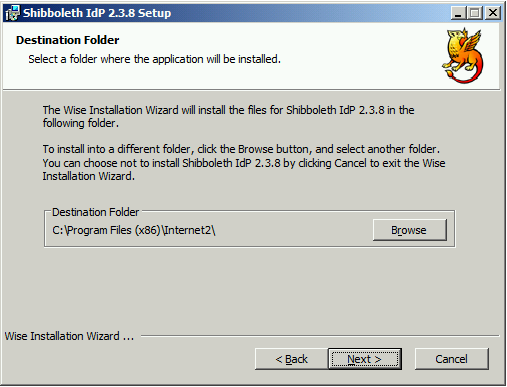
1. Click **Actions**. The **SmartScreen Filter – Windows Internet Explorer** opens up.



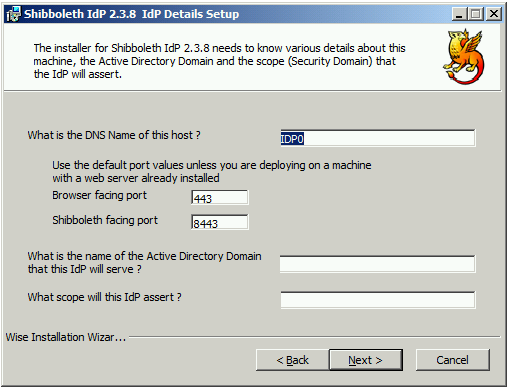
1. Click **More Options**, and then click **Run anyway**. The **Shibboleth IdP 2.3.8 Setup** wizard opens up.



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



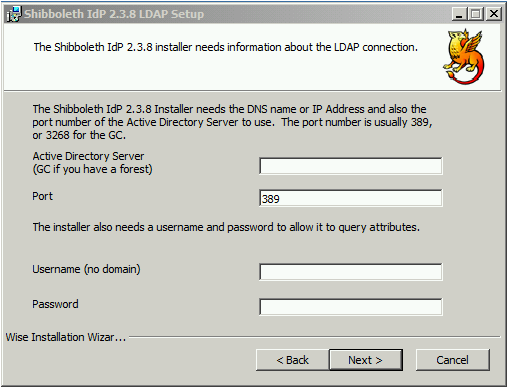
1. In the **Destination Folder** page, keep the defaults and then click **Next**.



1. If the installation package was run on a server that belongs to an Active Directory domain, i.e. a member server, the IdP Details Setup page of the wizard would have been automatically populated with information about the host Active Directory. Since the Shibboleth 2 IdP is installed on a standalone server with AD LDS as the LDAP directory to illustrate non-AD directories-based configuration, this has implications for the installation of the Shibboleth 2 IdP software.

In such case, this page and the next one have no suggested entries.

1. Type “*idp0.shib.idmgt.archims.fr*” in **What is the DNS Name of this host?** This value is used to name the Shibboleth 2 IdP endpoints in the generated metadata.
2. Type “*SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR*” in **What is the name of the Active Directory Domain that this IdP will serve ?** This is the name of the LDAP domain component from which the Shibboleth 2IdP will serve authentication and attributes. This value is primarily used to limit the search scope for LDAP lookups.
3. Leave **What scope will the IdP assert ?** empty
4. Click **Next**.



1. Type “*localhost*” in **Active Directory Server (GC if you have a forest)**. This value is used to generate the connection string for the LDAP connection to AD LDS.
2. Type “*CN=ShibSvc,CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR*” in **Username (no domain)**, and “*Password1*” in **Password**.
3. Click **Next**. The wizard proceeds with the installation.

****

1. Click **Finish** and minimize the *ReadMe.html* page.



**Note:**

*This installer automatically installs and configures* [*Apache Tomcat*](http://tomcat.apache.org/download-60.cgi)*[[84]](#footnote-84) 6.0.26 for 32-bit Windows. It depends on an existing 32-bit Java JRE installation.*

*The C:\Program File (x86)\Internet2\Shib2IdP* directory should now contains 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 2 IdP installation process generates a long-life self-signed certificate which is saved in this folder: the X.509 certificate (*idp.crt*), its private key (*idp.key*), and a related key store (*idp.jks*) containing both are put here.
* *lib*: contains the libraries (jars) that make up the Shibboleth 2 IdP. These are copies of those that occur in the IdP WAR file and are only used by the command line tools.
* *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-process log*: detailed description the IdP processing requests.
* *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.
* *Shib1Tools*: contains some JAR (Java ARchive) files such as log4j.
* *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 2 IdP.

*The C:\Program File (x86)\Internet2\Shib2IdPInstall* directory should now contains the *install.bat* batch file that enables you to regenerate the WAR file for any other reason.

The “captive” Tomcat 6 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 Tomcat’s file caching mechanisms.

The Shibboleth 2 documentation refers to this directory as IDP\_HOME.

Likewise, the *C:\Program File (x86)\Internet2\CaptiveTomcat 6.0* directory contains the “captive” Tomcat 6. The Shibboleth 2 documentation refers to this directory as TOMCAT\_HOME.

Set the TOMCAT\_HOME and IDP\_HOME system environment variable to their install directories. If you install to default locations, the right values to set will be:

TOMCAT\_HOME=C:\Program Files (x86)\Internet2\CaptiveTomcat 6.0

IDP\_HOME=C:\Program Files (x86)\Internet2\Shib2IdP

At this stage, the following system environment variables should be set:

|  |  |
| --- | --- |
| **Variable** | **Value name** |
| *JAVA\_HOME* | *C:\Program Files (x86)\Java\jre7* |
| *TOMCAT\_HOME* | *C:\Program Files (x86)\Internet2\CaptiveTomcat 6.0* |
| *IDP\_HOME* | *C:\Program Files (x86)\Internet2\Shib2IdP* |

### Installing the IdP ECP Extension on Shibboleth 2 versions prior to 2.3.3 (Optional)

If you’ve installed the Shibboleth 2 IdP with the Windows *shibboleth-identityprovider-2.3.8.msi* installation package as described in the previous section, skip this section and jump directly to the next one to configure the IdP. The enhanced client/proxy (ECP) extension is indeed included in Shibboleth 2.3.3 and later.

For earlier version of Shibboleth 2.x, you can download and install the Shibboleth Identity Provider ECP extension.

Though optional, deploying an ECP extension is a recommended step. If you are using Shibboleth as your STS, make sure to install such an extension in order for single sign-on to work with a smart phone, Microsoft Outlook or other clients.

This extension allows you to enable integration of desktop e-mail applications with Windows Azure AD. This extension implements the SAML 2.0 ECP specification. When you configure single sign-on with Windows Azure AD, you can specify the URL for the ECP extension, for example, *https://idp0.shib.idmgt.archims.fr/idp/profile/SAML2/SOAP/ECP*.

Section § 4.2.7.2 Enabling the Shibboleth ECP extension describes how to enable the Shibboleth ECP extension once installed.

**Note:**

*For more information, see the online help topic* [*IdP ECP Extension*](https://wiki.shibboleth.net/confluence/display/SHIB2/IdP+ECP+Extension)*[[85]](#footnote-85) on the Shibboleth Community wiki.*

### Configuring the Shibboleth 2 IdP

As previously outlined, the Shibboleth 2 IdP software must be configured once installed. The configuration typically requires to:

* Configure the SSL certificate for the IdP environment,
* Configure the SSL certificate for the “captive” Tomcat environment,
* Configure the users authentication for the IdP,
* Enable the LDAP 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 Windows 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 2 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 this two bullets will be instead specifically covered in section § 4.3 Configuring Shibboleth for use with single sign-on where we will setup the trust with the Windows 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 2 IdP software, see the online help topic* [*IdPConfiguration*](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPConfiguration)*[[86]](#footnote-86) on the Shibboleth Community wiki.*

The Shibboleth 2 IdP uses the following configuration files to control various aspects of its operation:

|  |  |
| --- | --- |
| **Configuration file** | **Description** |
| *attribute-filter.xml* | Configures the release of attributes to SP's. |
| *attribute-resolver.xml* | Configures attribute collection, transformation, and encoding. |
| *handler.xml* | Configures how the IdP receives and responds to various message types. |
| relying-party.xml | Configures how the IdP processes messages that are received. |
| *logging.xml* | Configuration of the IdP's logging system. You might want to use this to debug problems. |
| *login.config* | Configuration for the Username/Password authentication mechanism. |
| *service.xml* | Configuration for coarse grained IdP components. (Most people will never edit this). |
| *internal.xml* | Low-level IdP configuration file. (Most people will never edit this). |
| *tc\_config.xml* | Terracotta clustering configuration. |

These configuration files are at *%IDP\_HOME%\conf*, e.g. *C:\Program Files (x86)\Internet2\Shib2IdP\conf*.

Here are the files that need to be changed for the moment: *handler.xml* and *login.config* files.

**Note:**

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

#### Configuring the SSL certificate for Shibboleth

In section § 4.2.3 Requesting a SSL certificate, you’ve acquired a SSL certificate as a .pfx file (idp0.shib.idmgt.archims.fr.pfx) and save its content under the *C:\* directory as:

* The *idp0.shib.idmgt.archims.fr.crt* file for the SSL certificate,
* The *idp0.shib.idmgt.archims.fr.key* file for its private key,
* The *idp0.shib.idmgt.archims.fr.jks* file for both.

Description: Description: Description : Description : procedure_ddProceed with the following steps:

1. Copy the three above files to *%IDP\_Home%\credentials*, e.g. *C:\Program Files (x86)\Internet2\Shib2IdP\credentials*.
2. Use Windows Explorer to navigate to *%IDP\_Home%\conf*, e.g. *C:\Program Files (x86)\Internet2\Shib2IdP\conf*.
3. Right-click the *relying-party.xml* file, and then click **Edit**. The file should open in Notepad.
4. Press Ctrl+F to find “*Security Configurations*”.
5. Edit the <security:PrivateKey> and <security:Certificate> elements to refer to those files instead of the *idp.xxx* files that correspond to the long-life self-signed certificate.

This part of the *relying-party.xml* file should look like this:

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

<!-- Security Configurations -->

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

<security:Credential id="IdPCredential" xsi:type="security:X509Filesystem">

<security:PrivateKey>

C:\Program Files (x86)\Internet2\Shib2Idp/credentials/idp0.shib.idmgt.archims.fr.key

</security:PrivateKey>

<security:Certificate>

C:\Program Files (x86)\Internet2\Shib2Idp/credentials/idp0.shib.idmgt.archims.fr.crt

</security:Certificate>

</security:Credential>

1. Save and close the *relying-party.xml* file.
2. Use Windows Explorer to navigate to *%IDP\_Home%\metadata*, e.g. *C:\Program Files (x86)\Internet2\Shib2IdP\metadata*.
3. Right-click the *idp-metadata.xml* file, and then click **Edit**. The file should open in Notepad.
4. Delete the contents of both the *<X509Certificate>* elements and replacing them in each case with the base 64-encoded content of the SSL certificate file (not the private key file).
5. Open the *idp0.shib.idmgt.archims.fr.crt* file.
6. Copy the block of the base 64 encoded text in between the first set of -----BEGIN CERTIFICATE----- & -----END CERTIFICATE----- tags.

-----BEGIN CERTIFICATE-----   
(Block of base 64 encoded Text)   
-----END CERTIFICATE-----

1. Paste the block of the base 64 encoded text in both the <X509Certificate> elements.
2. Save and close the *idp-metadata.xml* file.

#### Configuring the SSL certificate for the Tomcat

The “captive” Tomcat has a connector configured in the Tomcat *%TOMCAT\_HOME%\conf\server.xml* file for each of ports 443 and 8443.

By default, the IdP long-life self-signed is also used to protect the IdP's endpoint URLs, and these are configured in the connectors in the *server.xml* file. So, we also need to modify these elements to take into account our SSL certificate.

Description: Description: Description : Description : procedure_ddTo declare our SSL certificate in the two connectors, proceed with the following steps:

1. Use Windows Explorer to navigate to *%TOMCAT\_Home%\conf*, e.g. *C:\Program Files (x86)\Internet2\CaptiveTomcat 6.0\conf*.
2. Right-click the *server.xml* file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “*<Connector port=8443*”.
4. Modify the *keystoreFile* parameter to declare the *idp0.shib.idmgt.archims.fr.jks* file.
5. Modify the *keystorePass* parameter to set the related password, for example “*changeit*” in our configuration.

This part of the *relying-party.xml* file should look like this:

<!--

One Connector on 8443 (for the SOAP connection). We delegate the

checking of the certificate to the IdP (which looks up the metadata).

You will usually not have to make any changes to this connector

-->

<Connector port="8443"

protocol="org.apache.coyote.http11.Http11Protocol"

maxHttpHeaderSize="8192"

maxSpareThreads="75"

scheme="https"

secure="true"

clientAuth="TRUE"

SSLEnabled="true"

sslProtocol="TLS"

SSLImplementation="edu.internet2.middleware.security.tomcat6.DelegateToApplicationJSSEImplementation"

keystoreFile="C:/Program Files (x86)/Internet2/Shib2IdP/credentials/idp0.shib.idmgt.archims.fr.jks"

keystorePass="changeit" />

1. Repeat steps 3 to 5 with “*<Connector port=443*”. This part of the relying-party.xml file should look like this:

<!--

This is the Connector that the browser speaks to (on port 443). You will

change the KeyStoreFile and KeyStorePass to be real CA signed ones

-->

<Connector port="443"

protocol="org.apache.coyote.http11.Http11Protocol"

maxHttpHeaderSize="8192"

maxSpareThreads="75"

scheme="https"

secure="true"

SSLEnabled="true"

sslProtocol="TLS"

keystoreFile="C:/Program Files (x86)/Internet2/Shib2IdP/credentials/idp0.shib.idmgt.archims.fr.jks"

keystorePass="changeit" />

1. Save and close the *idp-metadata.xml* 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.

This supposes to define a login handler (*<ph:LoginHandler>* element) in the %*IDP\_Home%\conf\handler.xml* file, which contains at least one authentication method (*<ph:AuthenticationMethod>* element).

**Note:**

*For more information, see the online help topic* [*IdPAuthUserPass*](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAuthUserPass)*[[87]](#footnote-87) on the Shibboleth Community wiki.*

Description: Description: Description : Description : procedure_ddProceed with the following steps:

1. Use Windows Explorer to navigate to *%IDP\_Home%\conf*, e.g. *C:\Program Files (x86)\Internet2\Shib2IdP\conf*.
2. Right-click the *handler.xml* file, and then click **Edit**. The file should open in Notepad.
3. Scroll until you see the following section:

<!-- Username/password login handler -->

1. Ensure the *<ph:LoginHandler>* element is set as follows:

<ph:LoginHandler xsi:type="ph:UsernamePassword"

jaasConfigurationLocation="file:///C:\Program Files (x86)\Internet2\Shib2Idp/conf/login.config">

<ph:AuthenticationMethod>

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

</ph:AuthenticationMethod>

<ph:AuthenticationMethod>

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

</ph:AuthenticationMethod>

</ph:LoginHandler>

1. If necessary, modify the *<ph:LoginHandler>* element to reflect the above settings. More particularly, check that the *jaasConfigurationLocation* correctly points to the location of the *login.config* file, which is included in the Shibboleth 2 IdP distribution (in the *%IDP\_Home%\conf* directory).
2. Save and close the *handler.xml* file.

#### Configuring Shibboleth LDAP authentication

The *login.config* file should already contain a Java Authentication and Authorization Service (JAAS) configuration to connect to our LDAP service using the LDAP authentication module shipped as part of the Shibboleth 2 IdP.

**Note:**

*For more information on JAAS configuration, see the information page* [*JAAS Tutorials*](http://docs.oracle.com/javase/1.5.0/docs/guide/security/jaas/tutorials/index.html)*[[88]](#footnote-88) on the Oracle Web site.*

Ensure it is uncommented and make changes appropriate to your local configuration if needed.

Description: Description: Description : Description : procedure_ddProceed with the following steps:

1. Use Windows Explorer to navigate to *%IDP\_Home%\conf*, e.g. *C:\Program Files (x86)\Internet2\Shib2IdP\conf*.
2. Right-click the *login.config* file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “*edu.vt.middleware.ldap.jaas.LdapLoginModule required*”.
4. Edit the parameters of the JAAS configuration (*edu.vt.middleware.ldap.jaas.LdapLoginModule*) to reflect the settings outlined hereafter. This *ldapUrl* parameter targets the *"localhost*" since the LDAP repository, e.g. AD LDS, is hosted on the machine.

This part of the *login.config* file should look like this:

ShibUserPassAuth {

// Example LDAP authentication

// See: https://spaces.internet2.edu/display/SHIB2/IdPAuthUserPass

edu.vt.middleware.ldap.jaas.LdapLoginModule required

ldapUrl="ldap://localhost:389"

baseDn="CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR"

bindDn="CN=ShibSvc,CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR"

bindCredential="Password1"

ssl="false"

tls=”false”

userFilter="cn={0}"

subtreeSearch="false";

**Note:**

*Replace “Password1” by your own password previously set in section § 4.2.4.3* Creating a Shibboleth Service account *for the Shibboleth service user.*

**Note:**

*For more information, see the online help topic* [*IdPAuthUserPass*](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAuthUserPass)*[[89]](#footnote-89) on the Shibboleth Community wiki.*

1. Save and close the *login.config* file.
2. Right-click the *attribute-resolver.xml* file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “*LDAP Connector*”.
4. Update the following LDAP data connector to the previously installed AD LDS instance to reflect the settings outlined hereafter. This part of the *attribute-resolver.xml* file should look like this:

<!-- LDAP Connector - just attach to the AD LDAP -->

<resolver:DataConnector id="myLDAP"

xsi:type="LDAPDirectory"

xmlns="urn:mace:shibboleth:2.0:resolver:dc"

useStartTLS="false"

ldapURL="ldap://localhost:389"

baseDN="CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR"

principal="CN=ShibSvc,CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR"

principalCredential="Password1">

<FilterTemplate>

<![CDATA[

(cn=$requestContext.principalName)

]]>

</FilterTemplate>

<!-- We rely on the uniqueness of the objectSid. But it is binary so we \*must\* make it so -->

<LDAPProperty name="java.naming.ldap.attributes.binary" value="objectSid"/>

**Note:**

*Replace “Password1” by your own password previously set in section § 4.2.4.3* Creating a Shibboleth Service account *for the Shibboleth service user.*

**Note:**

*For more information, see the online help topic* [*ResolverLDAPDataConnector*](https://wiki.shibboleth.net/confluence/display/SHIB2/ResolverLDAPDataConnector)*[[90]](#footnote-90) on the Shibboleth Community wiki.*

#### Enabling the Shibboleth ECP extension

This section supposes that the ECP extension has been installed as part of the Shibboleth 2 IdP setup (see section § 4.2.6 Installing the IdP ECP Extension on Shibboleth 2 versions prior to 2.3.3 (Optional)).

The first step consists in enabling in the “captive” Tomcat server the HTTP Basic authentication using accounts in our LDAP AD LDS instance. Please note that the Shibboleth ECP extension authentication is currently limited to HTTP Basic authentication.

Description: Description: Description : Description : procedure_ddTo enable the HTTP Basic authentication, proceed with the following steps:

1. Use Windows Explorer to navigate to *%TOMCAT\_HOME%\conf*, e.g. *C:\Program Files (x86)\Internet2\CaptiveTomcat 6.0\conf*.
2. Right-click the *server.xml* file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “*<Realm>*”.
4. Move to the line BELOW the <Realm> element and insert the following text:

<Realm className="org.apache.catalina.realm.JNDIRealm"

debug="99"

connectionURL="ldap://localhost:389"

authentication="simple"

referrals="follow"

connectionName=" CN=ShibSvc,CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR "

connectionPassword="Password1"

userSearch="(cn={0})"

userBase=" CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR "

userSubtree="false"

allRolesMode = "authOnly" />

**Note:**

*Replace “Shib” and “Password1” by the username and password of your choice.*

1. Save and close the *server.xml* file.

You must then edit the *web.xml* file in the IdP's source code (located under *C:\Program File (x86)\Shib2IdPInstall\src\main\webapp\WEB-INF*) to enable the ECP extension and rebuild a WAR file with the *install.bat* batch file.

To enable the ECP extension, proceed with the following steps:



1. Use Windows Explorer to navigate to *C:\Program File (x86)\Shib2IdPInstall\src\main\webapp\WEB-INF*.
2. Right-click the *web.xml* file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “*<!—Uncomment to use container managed authentication -->*”.
4. Move to the line BELOW and insert the following text:

<security-constraint>

<display-name>Shibboleth IdP</display-name>

<web-resource-collection>

<web-resource-name>ECP</web-resource-name>

<url-pattern>/profile/SAML2/SOAP/ECP</url-pattern>

<http-method>GET</http-method>

<http-method>POST</http-method>

</web-resource-collection>

<auth-constraint>

<role-name>\*</role-name>

</auth-constraint>

<user-data-constraint>

<transport-guarantee>CONFIDENTIAL</transport-guarantee>

</user-data-constraint>

</security-constraint>

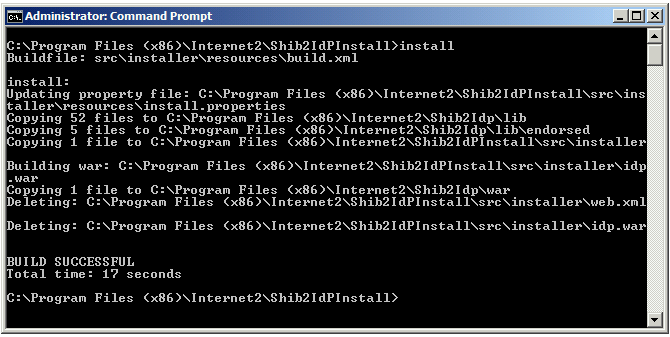
<login-config>

<auth-method>BASIC</auth-method>

<realm-name>ShibUserPassAuth</realm-name>

</login-config>

1. Save and close the *web.xml* file.
2. Open a command prompt and navigate to *C:\Program File (x86)\Shib2IdPInstall*.
3. Run the *install.bat* batch file to rebuild a WAR file for Shibboleth.



**Note:**

*For more information, see the online help topic* [*IdPEnableECP*](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPEnableECP)*[[91]](#footnote-91) on the Shibboleth Community wiki.*

#### Adjusting the level of logging

At the top of the *%IDP\_HOME%\conf\logging.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\logging.xml* may give the right level of information to start with the previous configuration:

<!-- Logs IdP, but not OpenSAML, messages -->

<logger name="edu.internet2.middleware.shibboleth" level ="WARN"/>

<!-- Logs OpenSAML, but not IdP, messages -->

<logger name="org.opensaml" level="WARN"/>

<!-- Logs LDAP related messages -->

<logger name="edu.vt.middleware.ldap" level="WARN"/>

<!-- Logs inbound and outbound protocols messages at DEBUG level -->

<logger name="PROTOCOL\_MESSAGE" level="TRACE"/>

<logger name="Shibboleth-Access" level="ALL" >

<appender-ref ref="IDP\_ACCESS" />

</logger>

<logger name="Shibboleth-Audit" level="ALL">

<appender-ref ref="IDP\_AUDIT" />

</logger>

<logger name="org.springframework" level value="OFF" />

<logger name="org.apache.catalina" level value="ERROR"/>

**Note:**

*For more information, see the online help topic* [*IdPLogging*](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPLogging)*[[92]](#footnote-92) on the Shibboleth Community wiki.*

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

Description: Description: Description : Description : procedure_ddProceed with the following steps:

1. Use Windows Explorer to navigate to *%IDP\_Home%\conf*, e.g. *C:\Program Files (x86)\Internet2\Shib2IdP\conf*.
2. Right-click the *logging.xml* file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “*edu.vt.middleware.ldap.jaas.LdapLoginModule required*”.
4. Set the **logger** elements to reflect the above configuration.
5. Save and close the *logging.xml* file.

#### Configuring a local user for Tomcat Manager access

Description: Description: Description : Description : procedure_ddProceed with the following steps:

1. Use Windows Explorer to navigate to *%TOMCAT\_HOME%\conf*, e.g. *C:\Program Files (x86)\Internet2\CaptiveTomcat 6.0\conf*.
2. Right-click the *tomcat-users.xml* file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “*</tomcat-users>*”.
4. Move to the line ABOVE and insert the following text:

<role rolename="manager-gui"/>

<user username="Shib"

password="Password1"

roles="manager-gui"/>

**Note:**

*Replace “Shib” and “Password1” by the username and password of your choice.*

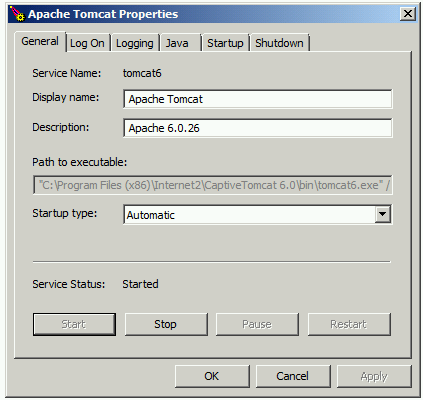
1. Save and close the *tomcat-users.xml* file.

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

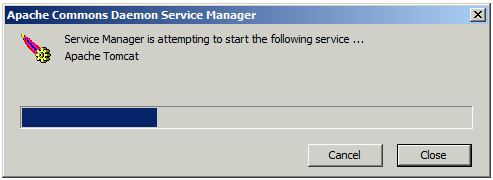
#### Restarting the Shibboleth Web server to take into account the updated configuration

Description : procedure_ddTo restart the Shibboleth Web server and check for start-up errors, proceed with the following steps:

1. Click **Start**, **All Programs**, **Manage Captive Tomcat**. The **Apache Tomcat Properties** dialog opens up.



1. On the **General** tab, click **Stop** near the bottom.
2. Use Windows Explorer to navigate to *%IDP\_Home%\logs*, e.g. *C:\Program Files (x86)\Internet2\Shib2IdP\logs*.
3. Delete all existing logs.
4. On the **Manage Captive Tomcat** interface, click **Start** near the bottom.



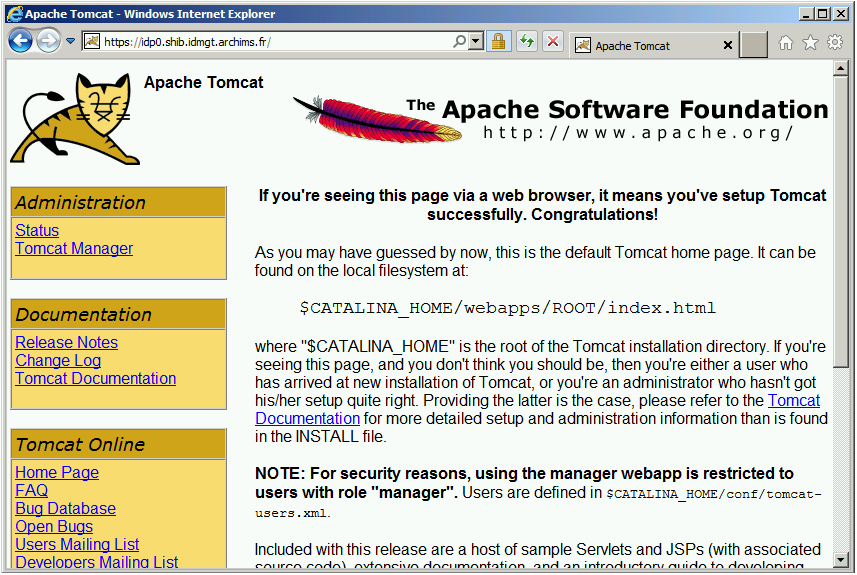
1. Check the files in *%IDP\_Home%\logs* for errors and search/troubleshoot as necessary. If you are still stumped, please check out [Shibboleth troubleshooting](https://spaces.internet2.edu/display/SHIB2/Troubleshooting)[[93]](#footnote-93) on the Shibboleth Community wiki.

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

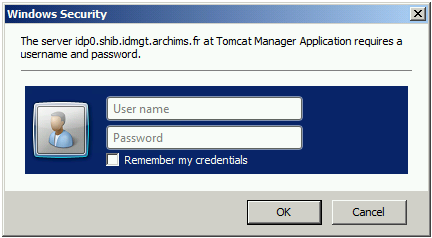
#### Ensuring that the Shibboleth IdP is running

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

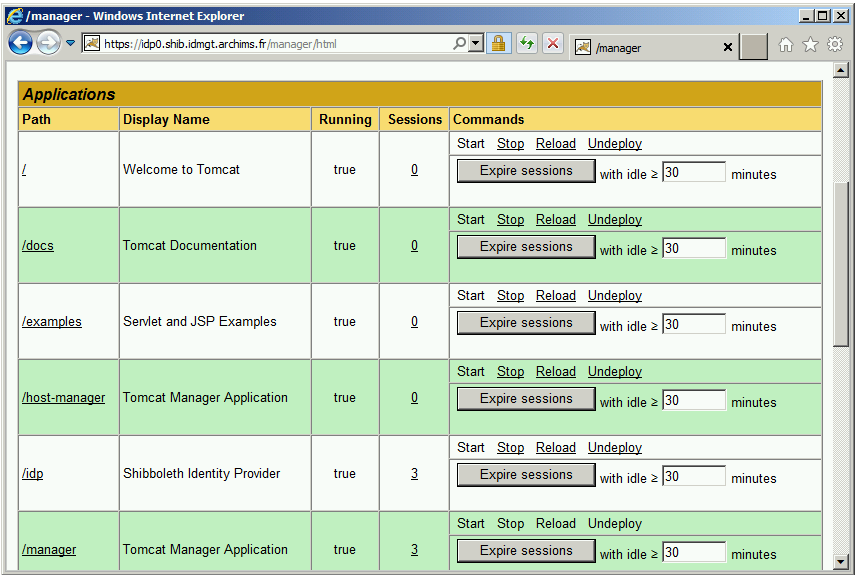
1. Launch Internet Explorer and browse to <https://idp0.shib.idmgt.archims.fr>.



1. Under **Administration**, click **Tomcat Manager**. A **Windows Security** dialog opens up.



1. Provide the credentials defined in section § 4.2.7.3 Configuring a local user for Tomcat Manager access, and then click **OK**.



1. Check the **Display Name** column for *Shibboleth Identity Provider* and ensure that the **Running** column equals *true*.

If not, you likely have an error in *%IDP\_Home%\logs*. If you are still stumped, please check out [Shibboleth troubleshooting](https://spaces.internet2.edu/display/SHIB2/Troubleshooting)[[94]](#footnote-94) at the Shibboleth Community wiki site.

#### Performing configuration tests

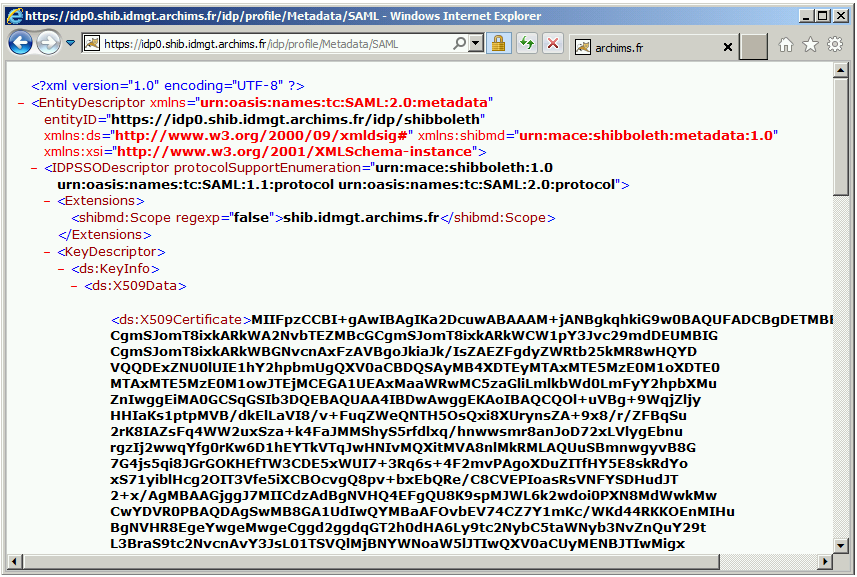
Description : procedure_ddProceed with the following steps:

1. Use Windows Explorer to navigate to *%IDP\_Home%\conf*, e.g. *C:\Program Files (x86)\Internet2\Shib2IdP\conf*.
2. Open the *ReadMe.html* file.

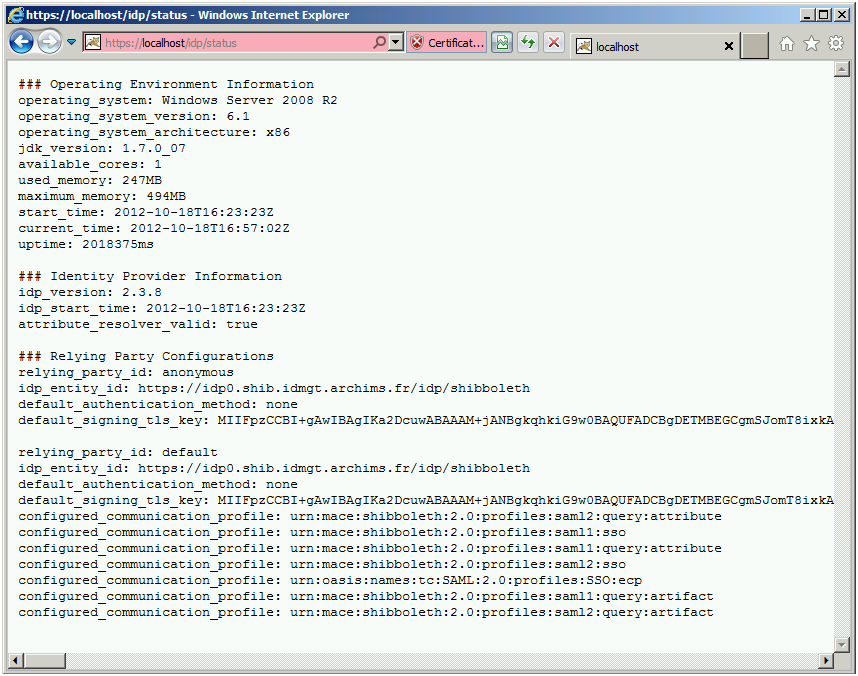


1. Attempt to visit the link for the Federation Metadata:

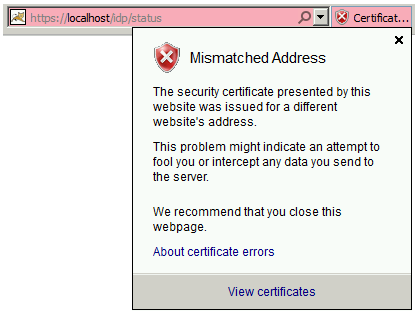
*“The metadata for your configuration is available here. You will need this information to register yourself with a Federation.”*

**

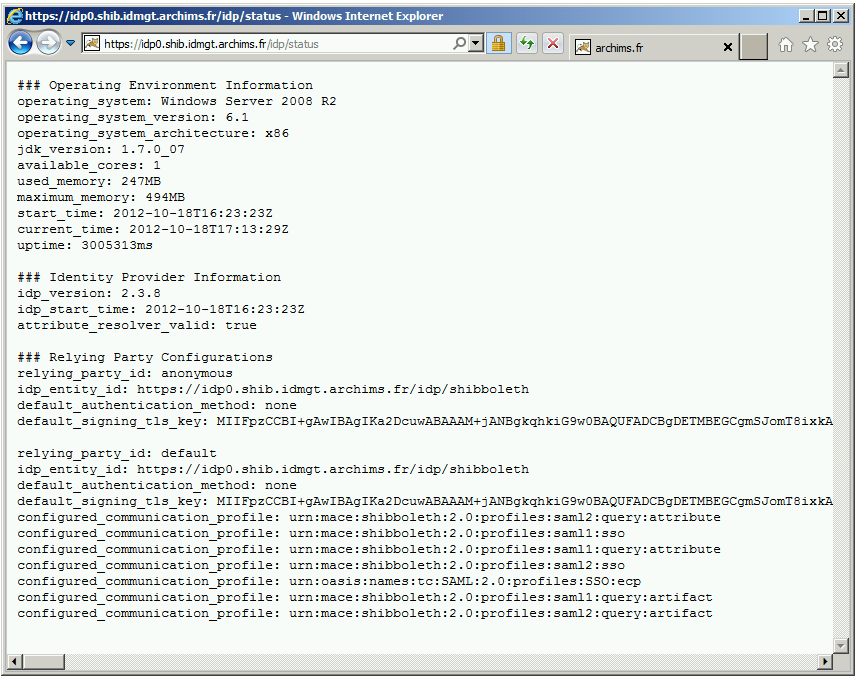
1. Attempt to visit the link for the IdP Status Page (<https://localhost/idp/status>). You should see a page like the following one. If this is the case, you can continue with the next section.



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



The *ReadMe.html* file indeed directs you to *localhost* instead of *idp0.shib.idmgt.archims.fr* because, by default, only *localhost* connection (those originating from 127.0.0.1) may access the page. As described, in the online help topic [IdPStatus](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPStatus)[[95]](#footnote-95) on the Shibboleth Community wiki site, you must edit the *web.xml* file in the IdP's source code (located under *C:\Program File (x86)\Shib2IdPInstall\src\main\webapp\WEB-INF*) to allow the host and rebuild a WAR file with the *install.bat* batch file.

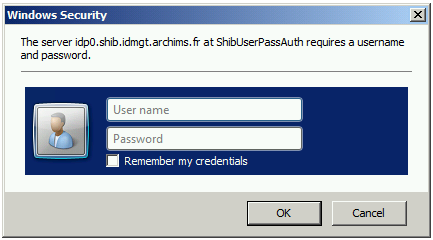


If you have installed (see section § 4.2.6 Installing the IdP ECP Extension on Shibboleth 2 versions prior to 2.3.3 (Optional)) and enabled (see section § 4.2.7.2 Enabling the Shibboleth ECP extension) the Shibboleth ECP extension, you can now test the HTTP Basic Authentication on the Shibboleth 2 IdP ECP endpoint.

To test the HTTP Basic Authentication, proceed with the following steps:



1. Navigate to the Shibboleth 2 IdP ECP endpoint at <https://idp0.shib.idmgt.archims.fr/idp/profile/SAML2/SOAP/ECP>. HTTP Basic Authentication should be required and a **Windows Security** dialog should open up.



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

*Username: User1*

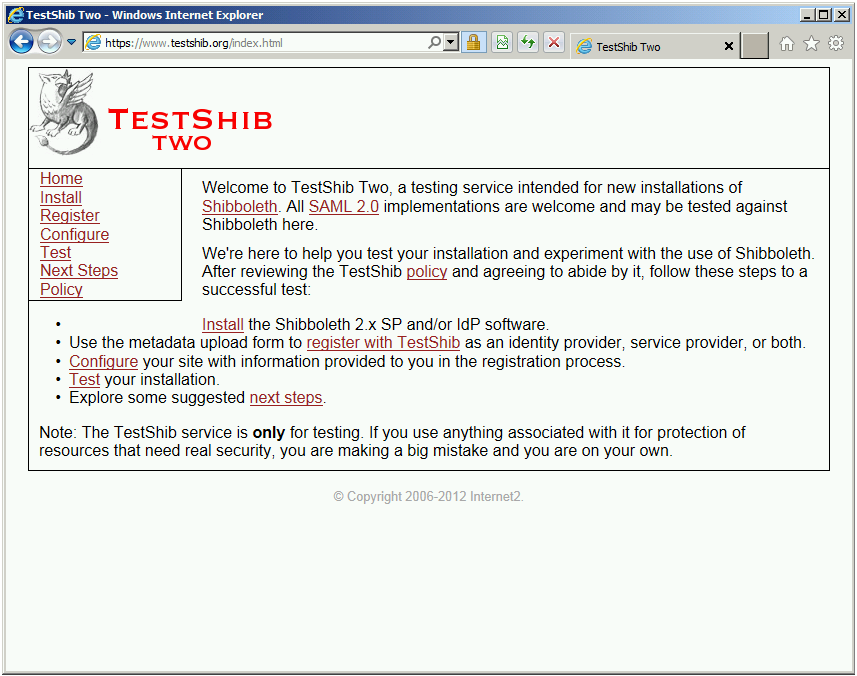
*Password: Password1*

1. Click **OK**. A HTTP 500 error or similar is expected after successful authentication.

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.

Description : procedure_ddTo (optionally) further test your Shibboleth 2 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 explains how to configure the Shibboleth IdP software deployed in the previous section to federate with Windows Azure AD using the OASIS SAML 2.0 protocol in order to enable single sign-on access to one or more Microsoft Online services such as Office 365. The SAML 2.0 relying party for a Microsoft cloud service like Office 365 illustrated in this paper is Windows Azure AD.

It describes what files are to be edited to appropriately configure the Shibboleth 2 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 2 IdP, for example in our case, *C:\Program Files (x86)\Internet2\Shib2IdP*. 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)*[[96]](#footnote-96).*

### Adding Windows Azure AD as a relying party

Adding a partner like Windows Azure AD into Shibboleth 2 consists in defining it in the %*IDP\_Home%\conf\relying-party.xml* file. Generally speaking, this file defines how the Shibboleth 2 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 in:

1. Adding a new relying party (*<rp:RelyingParty>* element) for the partner in order to enable the communication between Shibboleth 2 IdP and the partner, in our case Windows Azure AD;

**Note:**

*For information on the relying party, see the online help topic* [*IdPRelyingParty*](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPRelyingParty)*[[97]](#footnote-97) on the Shibboleth Community wiki site.*

1. Referencing the partner’s XML metadata document via a new metadata provider (*<MetadataProvider>* element).

Shibboleth 2 IdP indeed needs information about the Windows Azure AD relying party. Windows 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 2 IdP:

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

-or-

1. The file backed HTTP metadata provider:  configure Shibboleth 2 to pull the Windows 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* [*IdPMetadataProvider*](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPMetadataProvider)*[[98]](#footnote-98) on the Shibboleth Community wiki site.*

Description : procedure_ddTo save Windows Azure AD as a relying party, proceed with the following steps:

1. Use Windows Explorer to navigate to *%IDP\_Home%\conf*, e.g. *C:\Program Files (x86)\Internet2\Shib2IdP\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 “*</rp:DefaultRelyingParty>*”.
4. Move to the next line down and insert the following text.

<!—- Microsoft Windows Azure AD -->

<rp:RelyingParty id="urn:federation:MicrosoftOnline"

provider="https://idp0.shib.idmgt.archims.fr/idp/shibboleth"

defaultSigningCredentialRef="IdPCredential">

<rp:ProfileConfiguration xsi:type="saml:SAML2SSOProfile"

signAssertions="conditional"

encryptAssertions="never"

encryptNameIds="never" />

</rp:RelyingParty>

Make sure that:

* The relying party*id* value matches the *entityID* value of the *EntityDescriptor* element of the Windows Azure AD metadata, for example as of today "*urn:federation:MicrosoftOnline*";
* Your *provider* value matches your Shibboleth 2 IdP’s *entityID*specified in the *DefaultRelyingParty* element, for example in our case "*https://idp0.shib.idmgt.archims.fr/idp/shibboleth*".

The above configuration changes the default *saml:SAML2SSOProfile* settings in the *DefaultRelyingParty* element, which is required by Windows Azure AD to work.

<rp:ProfileConfiguration xsi:type="saml:SAML2SSOProfile" includeAttributeStatement="true"

assertionLifetime="PT5M" assertionProxyCount="0"

signResponses="never" signAssertions="always"

encryptAssertions="conditional" encryptNameIds="never"/>

1. You also have to specify the Shibboleth 2 IdP where to find the Windows Azure AD metadata document. As previously discussed, you can do this by adding another entry to the *relying-party.xml* file.
2. Still in Notepad with the *relying-party.xml* file opened, Press Ctrl+F to find “*</metadata:MetadataProvider>*”.

Move to the next line down and insert the following text to use the file backed HTTP metadata provider:

<!—- Microsoft Windows Azure AD Metadata -->

<MetadataProvider id="AAD"

xsi:type="FileBackedHTTPMetadataProvider"

xmlns="urn:mace:shibboleth:2.0:metadata"

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

backingFile="C:\Program Files (x86)\Internet2\Shib2IdP\metadata\AAD-FederationMetadata.xml">

<MetadataFilter xsi:type="ChainingFilter"

xmlns="urn:mace:shibboleth:2.0:metadata">

<MetadataFilter xsi:type="EntityRoleWhiteList">

<RetainedRole>samlmd:SPSSODescriptor</RetainedRole>

</MetadataFilter>

</MetadataFilter>

</MetadataProvider>

The file backed HTTP metadata provider loads the Windows 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 Windows Azure AD Metadata -->

<MetadataProvider id="AAD" xsi:type="ResourceBackedMetadataProvider">

<MetadataResource xsi:type="resource:FilesystemResource"

file=”C:\Program Files (x86)\Internet2\Shib2IdP\metadata\AAD-FederationMetadata.xml"/>

</MetadataProvider>

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

### Configuring the Shibboleth attribute resolver

Windows Azure AD requires two pieces of data from Shibboleth 2 IdP to locate the shadow account in the authentication platform.

1. **Windows Azure AD ImmutableID:** Windows Azure AD requires you select a unique identifier for each user in your user directory. You must also configure Shibboleth 2 to send this attribute on each federated login to Windows 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. The Windows Azure AD Service 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.archims.fr*.

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 § 4.2.4.4 Creating and configure a test user for the Shibboleth 2 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 or the user will not be able to login to the Microsoft Cloud services. For instance, with Active Directory, the DirSync tool automatically uses the Active Directory objectGUID for the ImmutableID value and processes the ImmutableID the same way. We mimic here the approach.

1. **Windows Azure AD UserID:** Windows Azure AD requires that the Windows Azure AD User ID, for example, *user@shib.idmgt.archims.fr*, 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 2 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* [*IdPAddAttribute*](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAddAttributeExamples)*[[99]](#footnote-99) on the Shibboleth Community wiki site.*

The Shibboleth 2 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*.

Description : procedure_ddTo 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)\Internet2\Shib2IdP\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 -->

<resolver:AttributeDefinition id="ImmutableID"

xsi:type="Simple"

xmlns="urn:mace:shibboleth:2.0:resolver:ad"

sourceAttributeID="uid">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="SAML2StringNameID"

xmlns="urn:mace:shibboleth:2.0:attribute:encoder"

nameFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" />

</resolver:AttributeDefinition>

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

<resolver:AttributeDefinition id="UserId"

xsi:type="ad:Simple"

sourceAttributeID="mail">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="IDPEmail"

friendlyName="UserId" />

</resolver:AttributeDefinition>

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

### Configuring the Shibboleth attribute filter

Shibboleth 2 IdP must be configured to release the two previous required attributes to Windows 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 (*<afp:AttributeFilterPolicy>* element) nodes that define rules (*<afp:PolicyRequirementRule>* element) for allowing a service provider like Windows 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* [*IdPAddAttributeFilter*](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAddAttributeFilter)*[[100]](#footnote-100) on the Shibboleth Community wiki site.*

Description : procedure_ddTo release the attributes to Windows Azure AD, proceed with the following steps:

1. Use Windows Explorer to navigate to *%IDP\_Home%\conf*, e.g. *C:\Program Files (x86)\Internet2\Shib2IdP\conf*.
2. Right-click the *attribute-resolver.xml* file, and then click **Edit**. The file should open in Notepad.
3. Press Ctrl+F to find “*urn:mace:shibboleth:2.0:afp:mf:saml classpath:/schema/shibboleth-2.0-afp-mf-saml.xsd">*”.
4. Move one line down and insert the following the text to modify the list of attributes that will be released:

<afp:AttributeFilterPolicy id="PolicyForWindowsAzureAD">

<afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString"

value="urn:federation:MicrosoftOnline" />

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

<afp:AttributeRule attributeID="UserId">

<afp:PermitValueRule xsi:type="basic:ANY" />

</afp:AttributeRule>

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

<afp:AttributeRule attributeID=ImmutableID">

<afp:PermitValueRule xsi:type="basic:ANY" />

</afp:AttributeRule>

<!-- Denying the transient ID release to Windows Azure AD -->

<afp:AttributeRule attributeID="transientId">

<afp:DenyValueRule xsi:type="basic:ANY"/>

</afp:AttributeRule>

</afp:AttributeFilterPolicy>

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

**The policy** **also denies for Windows Azure AD the release of the *transientId* which causes problems with the ECP endpoint.**

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

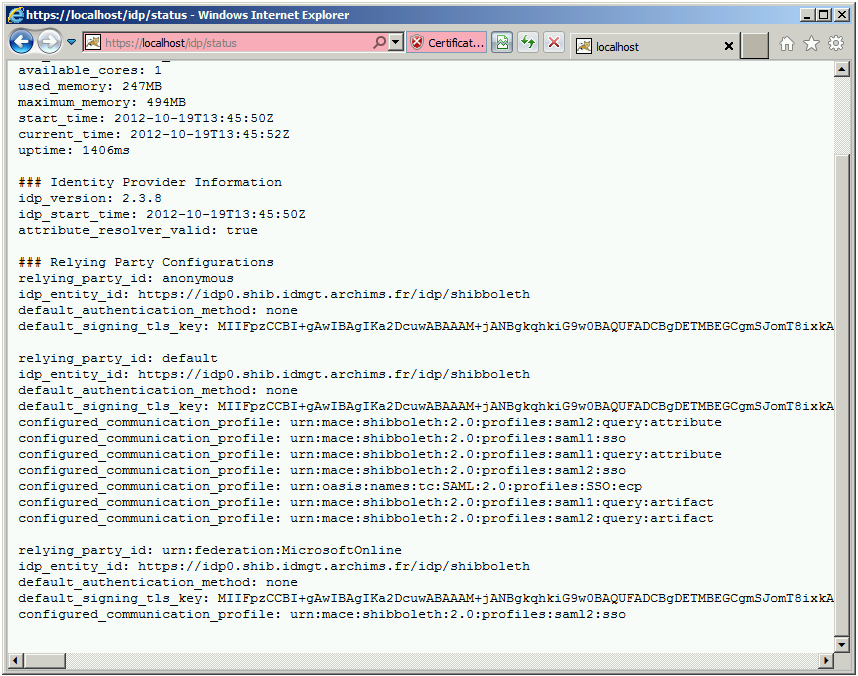
### Restarting Shibboleth 2 IdP and verifying functionality

Description : procedure_ddTo restart Shibboleth 2 IdP and verify functionality, proceed with the following steps:

1. Follow the steps as per section § 4.2.7.4 Restarting the Shibboleth Web server to take into account the updated configuration.
2. Follow the steps as per section § 4.2.7.5 Ensuring that the Shibboleth IdP is running.
3. Follow the steps as per section § 4.2.7.6 Performing configuration tests.

These steps stop and start Apache Tomcat to restart Shibboleth 2 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.

In the Shibboleth 2 IdP status page, Windows Azure AD should normally appear as a Relying Party at the bottom:



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

* *%TOMCAT\_HOME%\logs\*
  + *catalina.DATE.log*
  + *localhost.DATE.log*
* *%IDP\_HOME%\logs\*
  + *idp-process.log*

If you are still stumped, please check out [Shibboleth troubleshooting](https://spaces.internet2.edu/display/SHIB2/Troubleshooting)[[101]](#footnote-101) on the Shibboleth Community wiki.

## Installing Windows PowerShell for single sign-on with Shibboleth 2

Windows 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)*[[102]](#footnote-102).*

**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)*[[103]](#footnote-103), the* [*Windows PowerShell online help*](http://technet.microsoft.com/en-us/library/bb978526.aspx)*[[104]](#footnote-104), and the* [*Windows PowerShell Weblog*](http://blogs.msdn.com/powershell)*[[105]](#footnote-105)* [*Windows PowerShell Software Development Kit (SDK)*](http://msdn2.microsoft.com/en-us/library/aa830112.aspx)*[[106]](#footnote-106) that includes a programmer’s guide along with a full reference.*

To do this and to connect a Windows PowerShell command shell to the Windows Azure AD, you must have the required software for the Microsoft Online Services 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 (MOS SIA) 7.0;

Microsoft .NET Framework 3.51 0 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

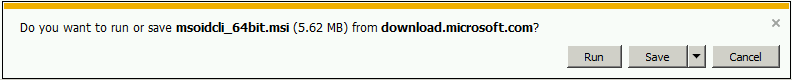
The Microsoft Online Services Sign-In Assistant (MOS SIA) 7.0 must be installed in order to use Microsoft Online Services Module for Windows PowerShell.

**Note:**

*The Microsoft Online Services Sign-In Assistant (MOS SIA) 7.0 provides end user sign-in capabilities to Microsoft Online Services, such Office 365 and Windows Azure AD Rights Management. In the context of this paper, the MOS 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 (MOS SIA)*](http://community.office365.com/en-us/w/office/534.aspx)*[[107]](#footnote-107).*

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

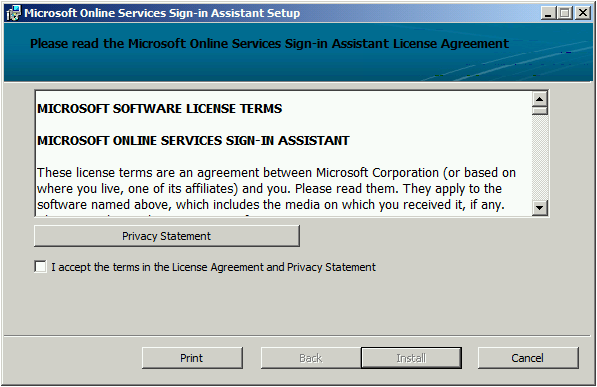
1. Download the Microsoft Online Services Sign-In Assistant (MOS SIA) 7.0 (*msoidcli\_64bit.msi*) from the following link: [Microsoft Online Services Sign-In Assistant for IT Professionals RTW](http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=28177)[[108]](#footnote-108) and click **Run** to install.



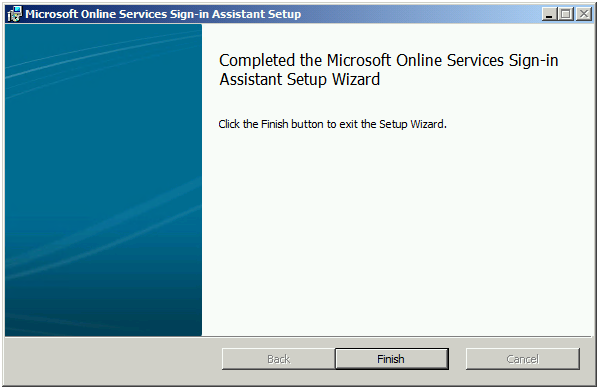
**Note:**

*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 MOS 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)*[[109]](#footnote-109).*

The wizard **Microsoft Online Services Sign-in Assistant Setup** starts.



1. Click **I accept the terms in the License Agreement and Privacy Statement** and click **Install**.



1. On the completion page, click **Finish**.

### Installing the Microsoft Online Services Module for Windows PowerShell

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

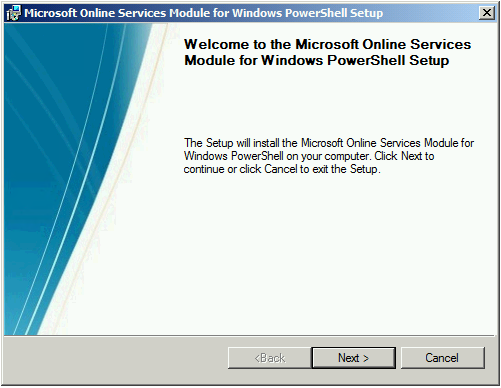
Administrative privileges are needed on the local computer in order to install the Microsoft Online Services Module.

Description : procedure_ddIn order to install the tool, proceed with the following steps:

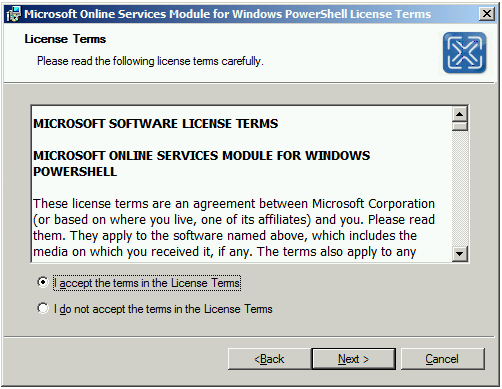
1. Download the Microsoft Online Services Module (*AdministrationConfig-en.msi*) from the following URL: [Microsoft Online Services Module for Windows PowerShell (64-bit version](http://go.microsoft.com/fwlink/p/?linkid=236293))[[110]](#footnote-110) and click **Run** to execute the setup the module.



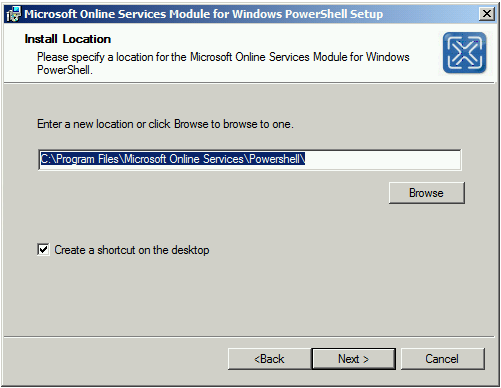
The **Microsoft Online Services Module for Windows PowerShell Setup** wizard starts.



1. On the **Welcome** page, select the **Next** option.



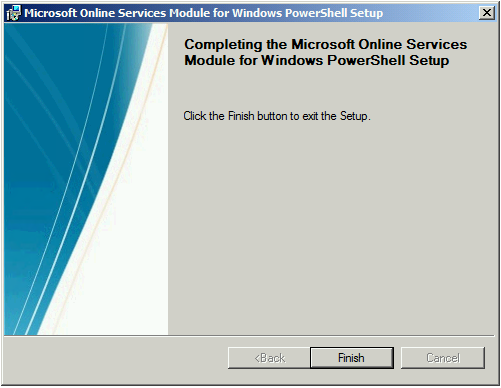
1. On the **License Terms** page, select **I accept the terms in the License terms** and click **Next**.



1. On the **Install Location** page, select the defaults for the installation location and click **Next**.



1. On the **Ready to Install** page, click **Install**.



1. On the completion page, click **Finish** option.

At this stage, the following cmdlets perform tasks related to single sign-on with Shibboleth 2.

**Note:**

*For more information about single sign-on cmdlets, see the Microsoft TechNet articles* [*Use Windows PowerShell cmdlets to manage your Windows Azure AD tenant*](http://technet.microsoft.com/en-us/library/jj151805)*[[111]](#footnote-111) and* [*Windows PowerShell cmdlet descriptions*](http://technet.microsoft.com/en-us/library/jj151835.aspx)*[[112]](#footnote-112).*

*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 Windows Azure AD

Windows 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 2 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 2 IdP and Windows Azure AD.

**Note:**

*For additional information, see the Microsoft TechNet article* [*Set up a trust between Shibboleth and Windows Azure AD*](http://technet.microsoft.com/en-us/library/jj205457)*[[113]](#footnote-113).*

### Connecting Windows PowerShell to the Microsoft Online Services

The next step is to open the Windows PowerShell from **Microsoft Online Services Module for Windows PowerShell** and connect the Windows PowerShell to the online domain using your Online Administrator Credentials.

Description : procedure_ddTo connect Windows PowerShell to the Microsoft Online Services, proceed as follows:

1. Click **Start** > **All Programs > Microsoft Online Services >** **Microsoft Online Services Module for Windows PowerShell** to open a Windows PowerShell command prompt with the single sign-on cmdlets.
2. From the Windows PowerShell command prompt, type **Connect-MsolService**. This command prompts for your Microsoft Online Admin credentials and sets the context of the Windows PowerShell as Online Administrator.

*Username: admin@idmgt14.onmicrosoft.com*

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



PS C:\Windows\system32> Connect-MsolService

**Note:**

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

### Creating the LDAP domain

After signing up for Office 365, the only domain associated with your subscription account is the onmicrosoft.com subdomain chosen during registration, e.g. *idmgt14.onmicrosoft.com* in our configuration.

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

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

1. Connect Windows PowerShell to the Microsoft Online Services (see eponym section § 4.5.1 Connecting Windows PowerShell to the Microsoft Online Services).

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.

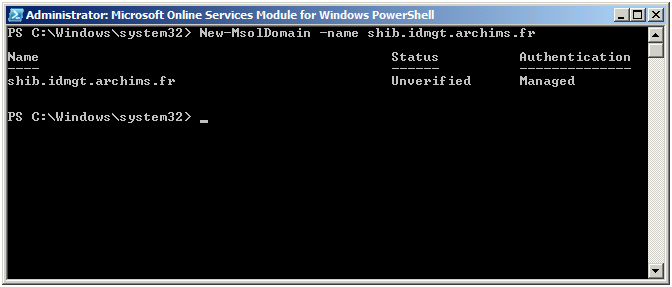
1. Create a new standard domain for the LDAP *shib.idmgt.archims.fr* domain with the following command:

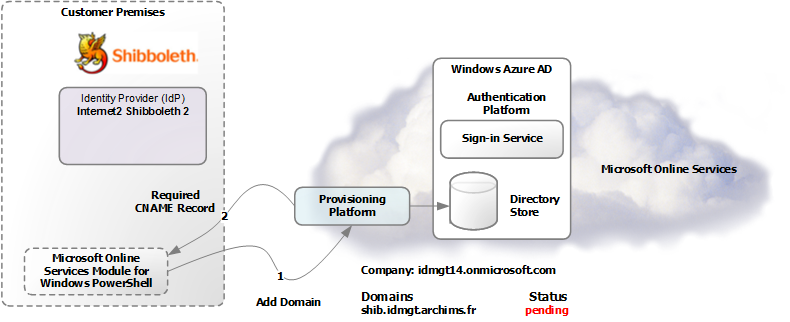
PS C:\Windows\system32> New-MsolDomain –name shib.idmgt.archims.fr

Name Status Authentication

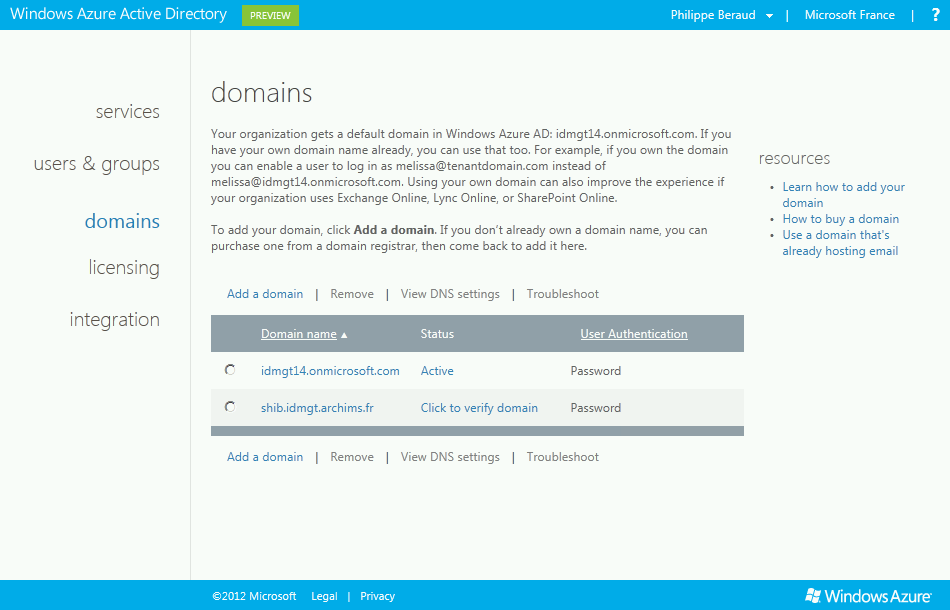
---- ------ --------------

shib.idmgt.archims.fr unverified Managed





The following snapshot illustrates the status of the newly created domain in the Windows Azure AD Management Portal.



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.archims.fr

canonicalName : ps.microsoftonline.com

ExtensionData : System.Runtime.Serialization.ExtensionDataObject

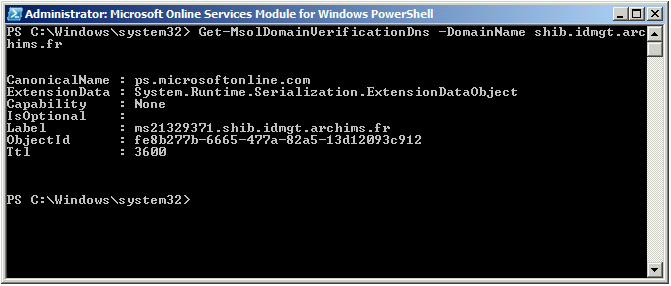
Capability : None

IsOptional :

Label : ms21329371.shib.idmgt.archims.fr

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 outputted above:

*Name: ms21329371.shib.idmgt.archims.fr*

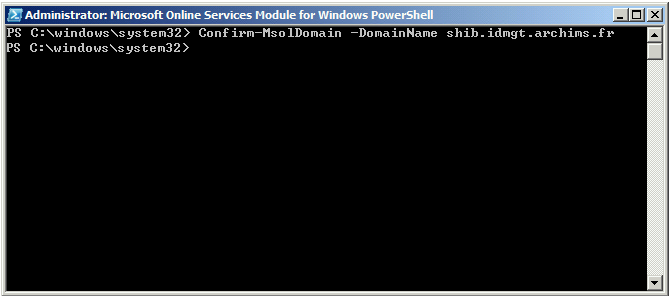
*Type: CNAME*

*Value: ps.microsoftonline.com*

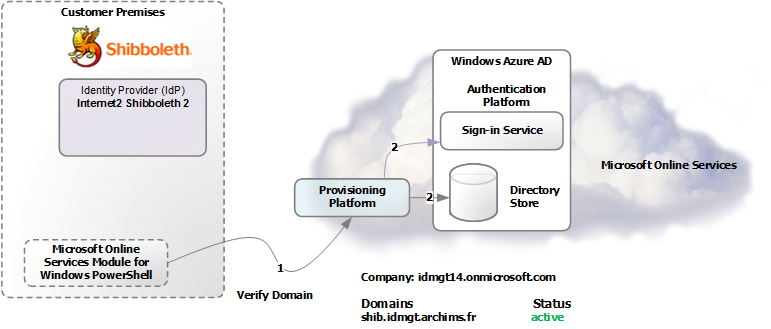
Windows Azure AD indeed uses a DNS record that you create at your registrar to confirm that you own the domain. For additional information, please refer to the Microsoft TechNet articles [Add your domain](http://technet.microsoft.com/en-us/library/hh969247.aspx)[[114]](#footnote-114) and [Verify a domain at any domain name registrar](http://technet.microsoft.com/en-us/library/jj151803.aspx)[[115]](#footnote-115).

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

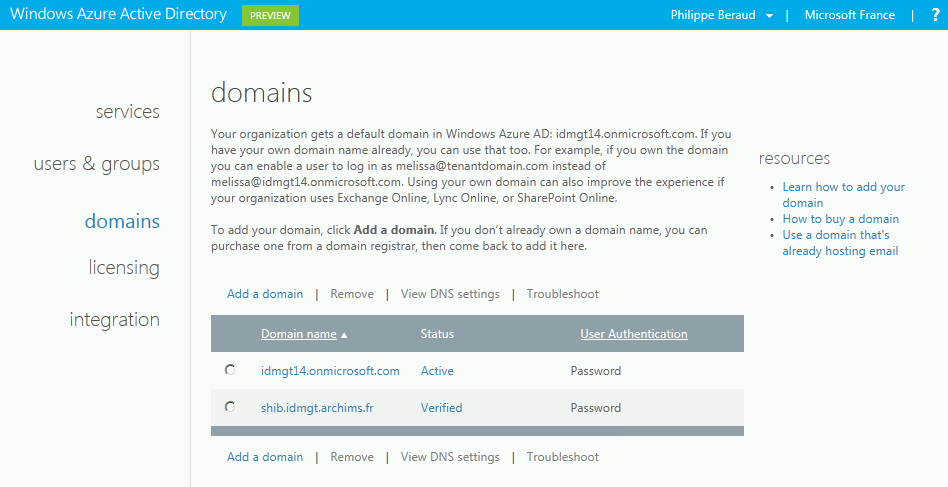
PS C:\Windows\system32> Confirm-MsolDomain –DomainName shib.idmgt.archims.fr



This verifies the domain proof of ownership.



The following snapshot illustrates the status of the newly created domain in the Windows Azure AD Management Portal.



**Note:**

*The (shib.idmgt.archims.fr) domain can also be added from the Microsoft Online Portal (MOP) or the Windows Azure AD Management Portal as well. The steps would be identical and the domain would still have to be verified in the same way.*

### 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 2 IdP and Windows 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 user 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 Windows 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 *RelyingParty* element added for Windows Azure AD in the *%IDP\_HOME%\conf\relying-party.xml* file (see section § 4.3.2 Adding Windows Azure AD as a relying party).

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. *shib.idmgt.archims.fr* in our configuration, as a federated domain.

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

1. Connect Windows PowerShell to the Microsoft Online Services (see eponym section § 4.5.1 Connecting Windows PowerShell to the Microsoft Online Services).

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.archims.fr*) for single sign on:

PS C:\Windows\system32> $dom = "shib.idmgt.archims.fr”

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

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

PS C:\Windows\system32> $ecpUrl = "https://idp0.shib.idmgt.archims.fr/idp/profile/SAML2/SOAP/ECP"

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

PS C:\Windows\system32> $logoutUrl = “https://idp0.shib.idmgt.archims.fr/logout”

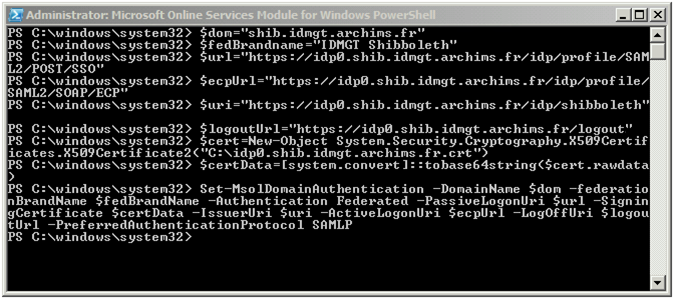
PS C:\Windows\system32> $cert = New-Object

System.Security.Cryptography.X509Certificates.X509Certificate2(

"idp0.shib.idmgt.archims.fr.crt")

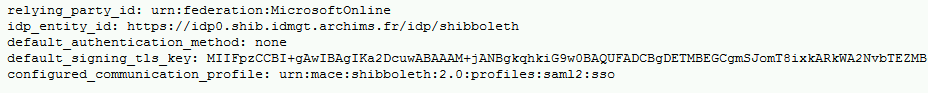
PS C:\Windows\system32> $certData = [system.convert]::tobase64string($cert.rawdata)

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



**Note:**

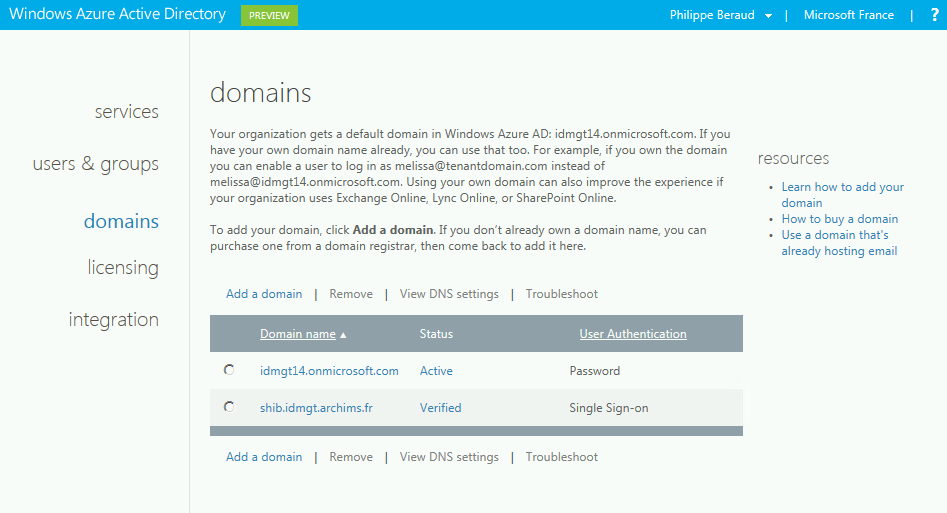
*The $certData can directly source from the Shibboleth 2 IdP status page. It indeed corresponds to the base64 encoded value of the default\_signing\_tls\_key parameter for the relying\_party\_id urn:federation:MicrosoftOnline, for example “MIIFpzCCBI+g…oam1ac=”.*



**Note:**

*You must run the command $ecpUrl = https://idp0.shib.idmgt.archims.fr/idp/profile/SAML2/SOAP/ECP only if you set up the Shibboleth Identity Provider ECP extension. Though an optional step, it is recommended that you install the Shibboleth Identity Provider ECP extension in order for single sign-on to work with a smart phone, Microsoft Outlook or other clients. For more information, see section § 4.2.6 Installing the IdP ECP Extension on Shibboleth 2 versions prior to 2.3.3 (Optional).*

After the above steps are completed, you can verify that the domain was added correctly and is federated via the Microsoft Online Portal (MOP). When you are in MOP, 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 § 1.2 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)[[116]](#footnote-116) *in the Windows Azure AD online documentation.*

We instead used Windows PowerShell cmdlets hereafter to provision our test user created in section § 4.2.4.4 Creating and configure a test user for the Shibboleth 2 IdP.

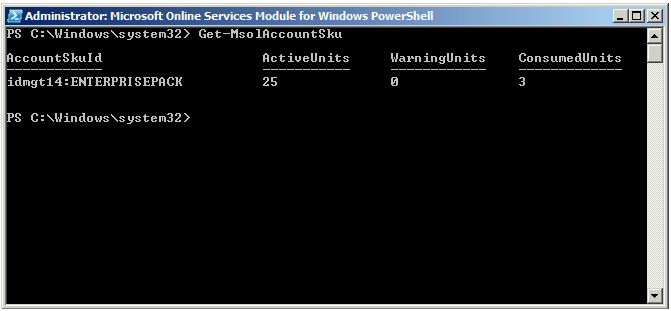
**This approach is not suitable for a production environment. Provisioning and synchronization are indeed not the same. Whilst with provisioning, you simply create objects and/or associated resources in a directory or external system, here Windows Azure AD, the synchronization integrates the provisioning 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 Windows Azure AD federated users from the Microsoft Online Services Module for Windows PowerShell.

Description : procedure_ddTo create a federated user, proceed with the following steps:

1. Connect Windows PowerShell to the Microsoft Online Services (see eponym section § 4.5.1 Connecting Windows PowerShell to the Microsoft Online Services).
2. Run the following command to obtain the unique string ID of the account/SKU combination that will be needed to assign a license, for example “*idmgt14:ENTERPRISEPACK*” in our configuration.

PS C:\Windows\system32> Get-MsolAccountSku

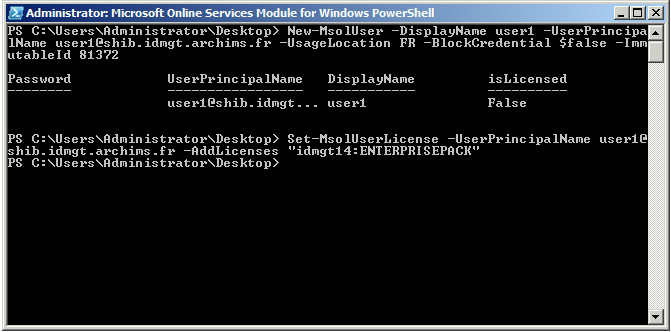


1. Create the user with the following command :

PS C:\Windows\system32> New-MsolUser -DisplayName user1 –UserPrincipalName user1@shib.idmgt.archims.fr -UsageLocation FR -BlockCredential $false –ImmutableId 81372

1. Set the user’s license with the following command:

PS: C:\Windows\system32> Set-MsolUserLicense -UserPrincipalName user1@shib.idmgt.archims.fr -AddLicenses "idmgt14:ENTERPRISEPACK"



You can now use this user to verify the single sign-on with Shibboleth 2.

## Verifying single sign-on with Shibboleth 2

As suggested in the Microsoft TechNet article [Verify single sign-on with Shibboleth](http://technet.microsoft.com/en-us/library/jj205458)[[117]](#footnote-117), it is always best 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 best just accessing the Microsoft Online Portal (MOP) or the Windows Azure AD 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 MOP but cannot log into OWA with the corporate credentials then you can be sure the issue is not related to SSO.

Description: Description : procedure_ddTo verify the SSO with the Shibboleth 2 IdP, proceed as follows:

1. Open a browser, and then navigate to <https://activedirectory.windowsazure.com> to access the Windows Azure AD Management 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.

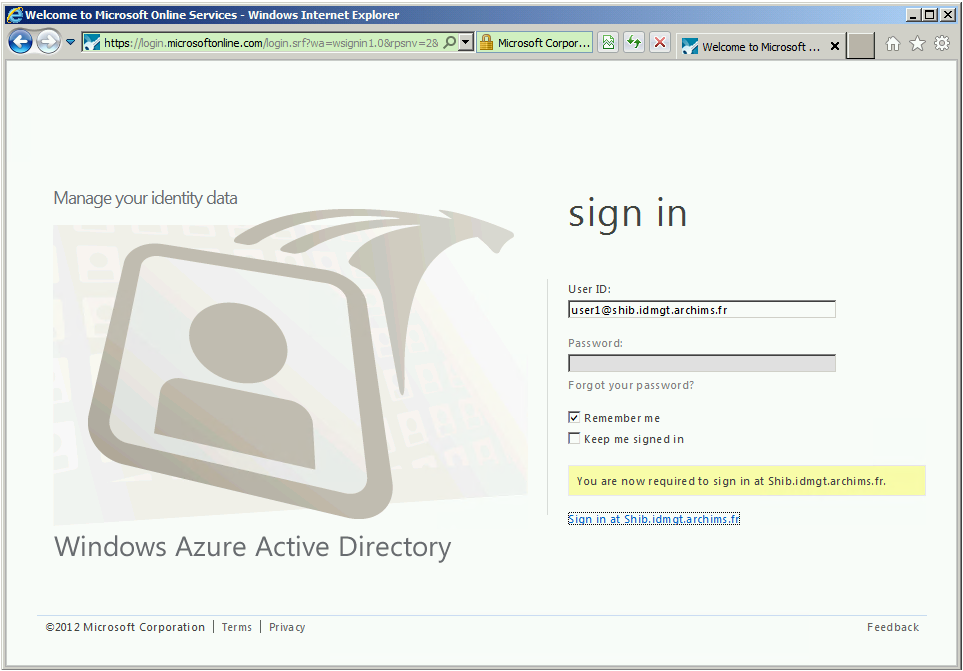
*Username: user1@shib.idmgt.archims.fr*

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 federated.

**Note:**

*If you turn on HTTP tracing on Internet Explorer or observe the traffic via a tool like the* [*Fiddler2*](http://www.fiddler.com)*[[118]](#footnote-118) 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 there results show the Shibboleth 2 IdP passive endpoint information.*

*Likewise, with Firefox, you can use the* [*SAML tracer*](https://addons.mozilla.org/en-US/firefox/addon/saml-tracer/)*[[119]](#footnote-119), 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 here password. **Password** will be shaded. The user is provided a link to the Shibboleth 2 IdP. You will see the following message: “*You are now required to sign at <your company>*”.
2. Click the **Sign in at <your company>** link, i.e. the **Sign in at Shib.idmgt.archims.fr** link from the above screen capture. This is a redirect link to send the user to the Shibboleth 2.0 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*

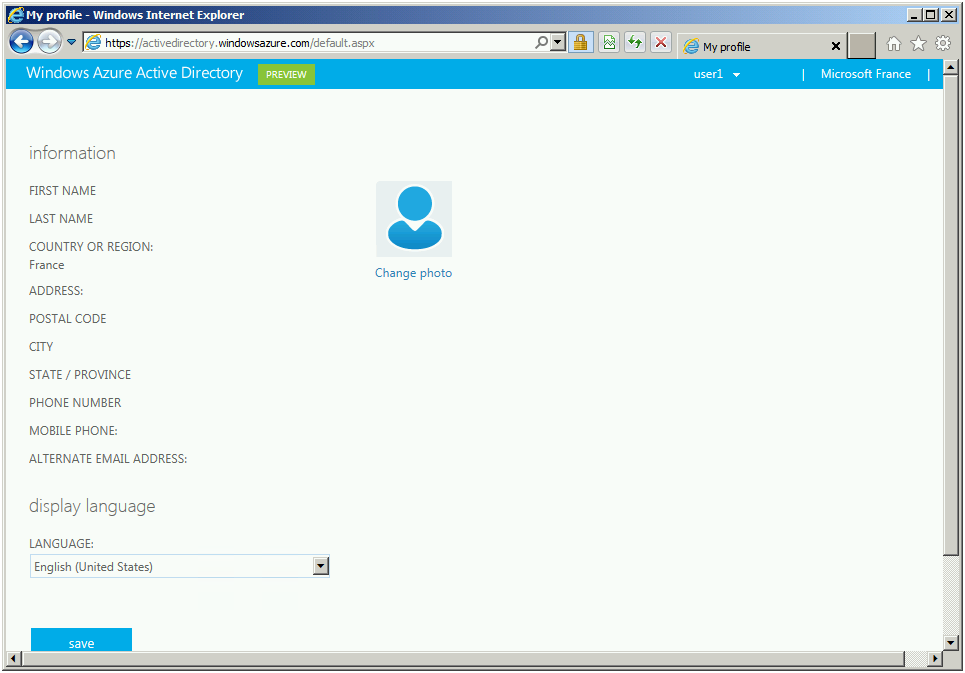
**Note:**

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

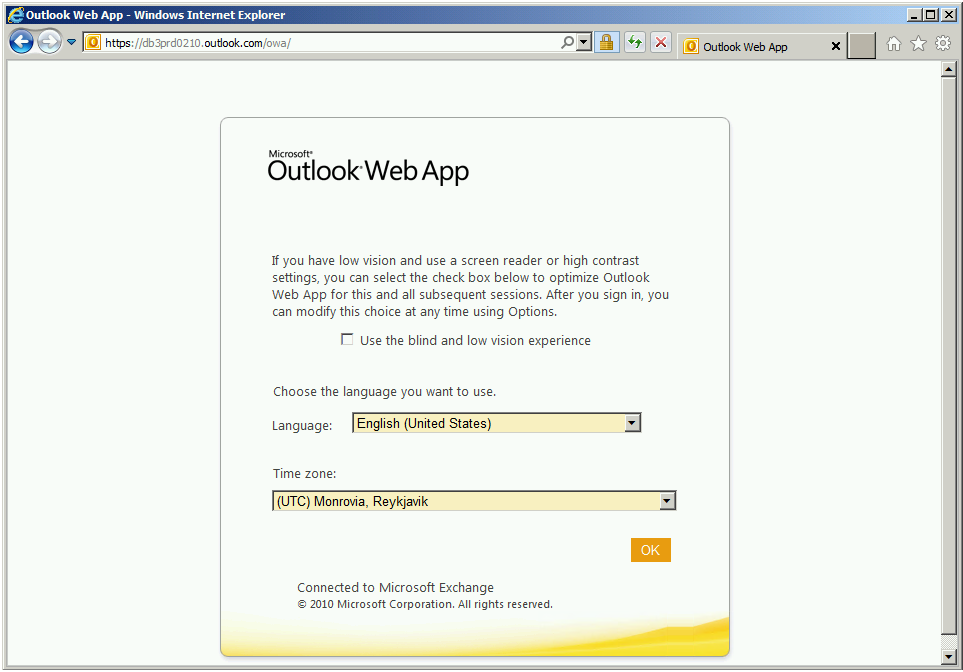
**Note:**

*To customize the default JSP (JavaServer Pages) login page, see the online help topic* [*IdPAuthUserPassLoginPage*](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAuthUserPassLoginPage)*[[120]](#footnote-120) on the Shibboleth Community wiki.*

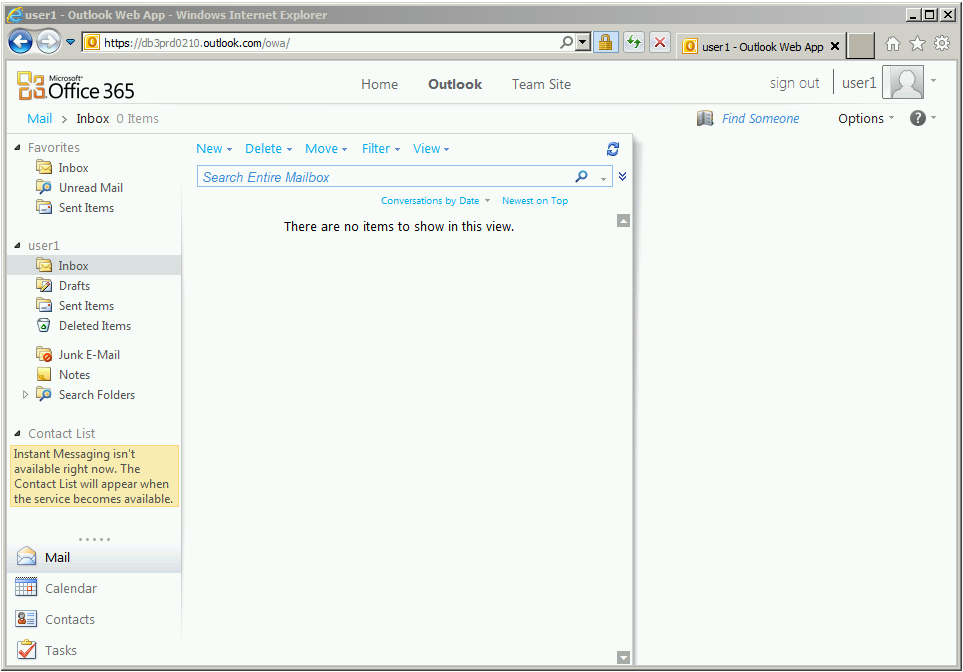
If you are able to sign in, the single sign-on has been set up correctly.



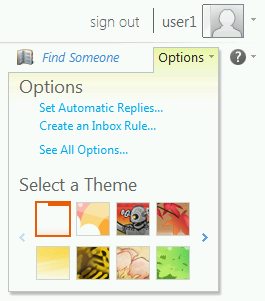
1. Navigate to the user mailbox at <http://outlook.com/idmgt14.onmicrosoft.com>.



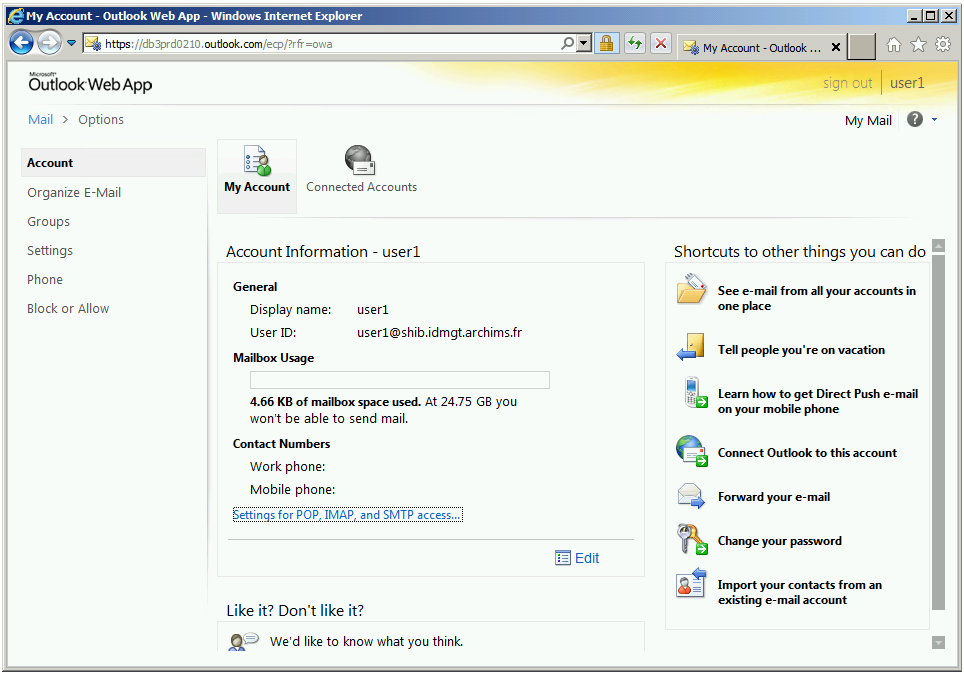
1. On the first use, you are invited to set your language and time zone. Click **OK** once done. You can now work with OWA.



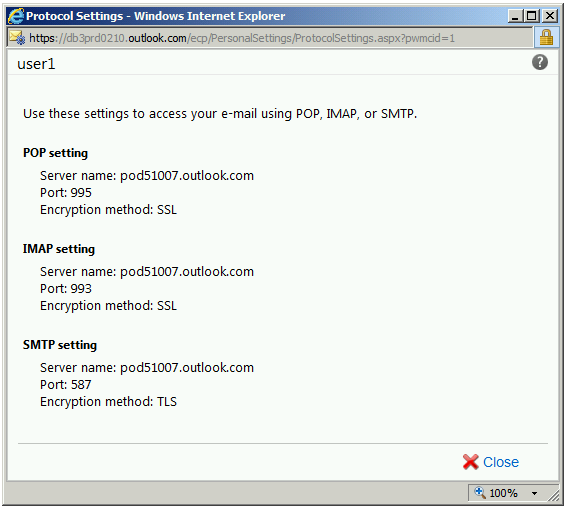
1. Click **Options** underneath user1 in the upper right corner.



1. Click **See All Options…**



1. Click **Settings for POP, IMAP, and SMTP access…** A **Protocol Settings** dialog opens up.

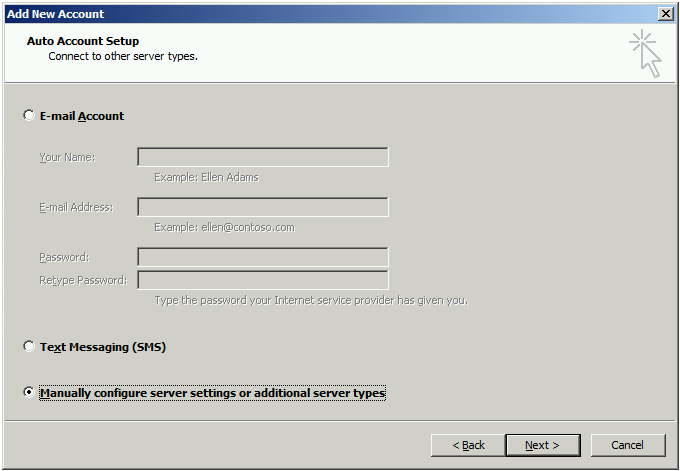


1. Memorize the IMAP and SMTP settings in order to appropriately configure your email rich client application.

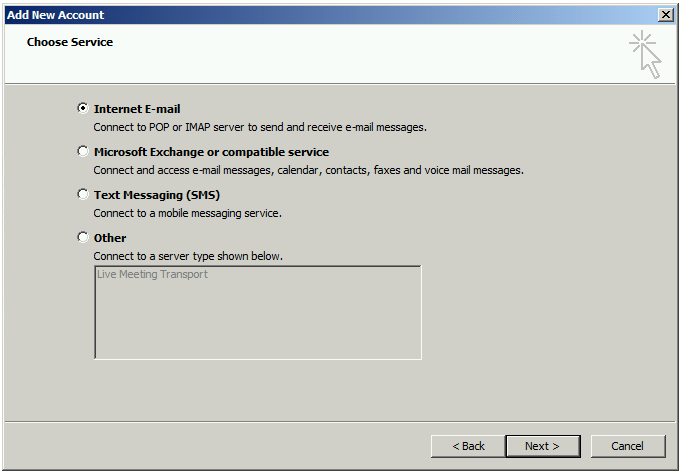
For that purpose, we illustrate hereafter the configuration of Outlook 2010 so that it can leverage the ECP extension via the Windows Azure AD organization’s tenant configuration.

Description: Description : procedure_ddTo configure Outlook 2010, proceed as follows:

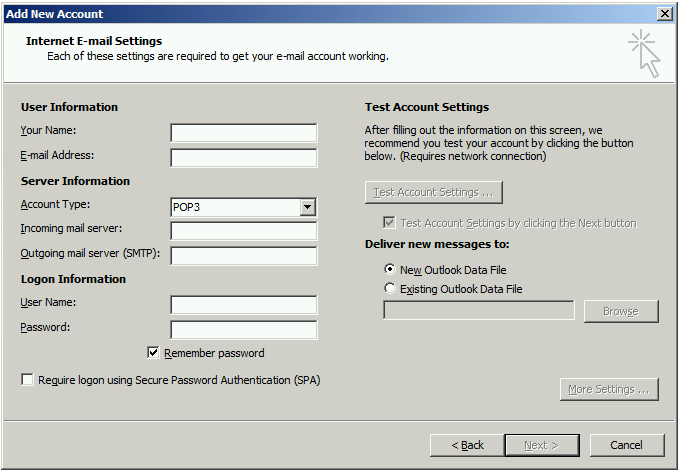
1. Launch Outlook 2010. When this is the first time Outlook 2010 is launched on the local computer, the **Microsoft Outlook 2010 Startup** wizard start and you will be prompted to setup an e-mail account. In that case, you can proceed with that wizard and configure the e-mail settings to use Exchange Online with the (above) tenant user account elements.
   * If the **Microsoft Outlook 2010 Startup** wizard is automatically launched, then on the first page, click **Next**. On the **E-mail Accounts** page of the wizard, click **Next** and you will be presented with the **Auto-Account Setup** page.
   * If Microsoft Outlook 2010 has already been used, then the Startup wizard will not appear. The account settings configuration must be launched manually. To do so, click the **File** menu in the Outlook 2010 toolbar. Then, just above the **Account Settings** button, click **Add Account** which will open the **Account Setup** page



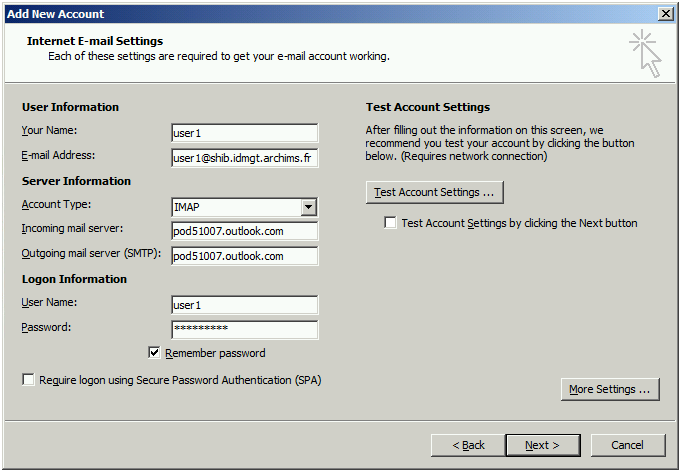
1. On the **Auto Account Setup** page, select **Manually configure server settings or additional server types** (in order to illustrate how to use the above settings in other possible contexts than Outlook 2010) and click **Next**.



1. Select **Internet E-mail** and click **Next**.



1. In **User Information**, type “*user1*” in **Your name** and “*user1@shib.idmgt.archims.fr*” in **E-mail Address**.
2. In **Server Information**, select **IMAP** in **Account Type** and type for example in our configuration “*pod51007.outlook.com*” in **Incoming mail server** and **Outgoing mail server (SMTP)**.
3. In **Logon Information**, type “*user1*” in **User Name** and “*Password1*” in **Password**. The page should look like the following one.



1. Click **Next** and then **Finish**. You should now have access to your mailbox via the Shibboleth ECP extension.

Beyond the above verification (and additional configuration), and depending on your on-premises environment and the identity directories used, you may test the following sign-in scenarios to ensure that single sign-on (SSO) using the Shibboleth 2 IdP is correctly configured and worked as expected.

You should test the access to the cloud services like Office 365 from browsers as well as rich client applications, such as Microsoft Office 2010, in the following environments (if applicable):

* From a domain-joined computer;
* From a non-domain-joined computer inside the corporate network;
* From a roaming domain-joined computer outside the corporate network;
* From the different operating systems that you use in your company;
* From a home computer;
* From an Internet kiosk (test access to the cloud service through a browser only);
* From a smart phone (for example, a smart phone that uses Microsoft Exchange ActiveSync (EAS)).

## Troubleshooting the single sign-on (SSO) with Shibboleth 2

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

If you are still stumped, please check out [Shibboleth troubleshooting](https://spaces.internet2.edu/display/SHIB2/Troubleshooting)[[121]](#footnote-121) 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)[[122]](#footnote-122).

# Understanding how federated authentication works in Office 365

This section aims at providing additional information on the configuration (via the modification of the Shibboleth 2 IdP configuration files and the use of the Microsoft Online Services Module for Windows PowerShell) in order to setup single sign-on (SSO). 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-premise Shibboleth infrastructure, Windows Azure AD (and its sign-in service), and the Microsoft Cloud services such as Office 365.

## Understanding the Shibboleth 2 configuration

As covered in section $ 4.3.1 Adding Windows Azure AD as a relying party, adding a partner like Windows Azure AD into a Shibboleth 2 IdP is mainly done by referencing the partner’s XML metadata.

Windows 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 Windows Azure AD metadata. Here is the current value of the metadata:

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

[<EntityDescriptor entityID="urn:federation:MicrosoftOnline" xmlns="urn:oasis:names:tc:SAML:2.0:metadata">](https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml)

[<SPSSODescriptor WantAssertionsSigned="true"](https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml)

[protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">](https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml)

<AssertionConsumerService Location="https://login.microsoftonline.com/login.srf"

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

index="0"

isDefault="true"/>

<AssertionConsumerService Location="https://login.microsoftonline.com/login.srf"

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

index="1"/>

<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>

</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)[[123]](#footnote-123) (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 the Windows Azure AD sign-in service in which we are interested.

This element defines two SAML 2.0 based endpoints for the Windows Azure AD sign-in service for the interaction with the organization’s on-premise 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 § 5.2 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 § 5.3 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 Windows Azure as a relying party in the Shibboleth 2 IdP configuration. The configuration metadata is loaded into the Shibboleth 2 IdP by metadata providers. Metadata providers are configured in the *relying-party.xml* file, which defines how the Shibboleth 2 IdP should interact with service providers such as Windows 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 Windows 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 Windows Azure AD platform side. Such a capability greatly lessens the administrative effort to maintain the relying party trust on the Shibboleth 2 IdP side.

In turn, as described in section § 2.4 Federation metadata defined, the Shibboleth 2 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.archims.fr/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.archims.fr

</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.archims.fr:8443/idp/profile/SAML1/SOAP/ArtifactResolution"

index="1"/>

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

Location="https://idp0.shib.idmgt.archims.fr: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.archims.fr/idp/profile/Shibboleth/SSO"/>

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

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

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

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

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

Location="https://idp0.shib.idmgt.archims.fr/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.archims.fr

</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.archims.fr:8443/idp/profile/SAML1/SOAP/AttributeQuery"/>

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

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

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

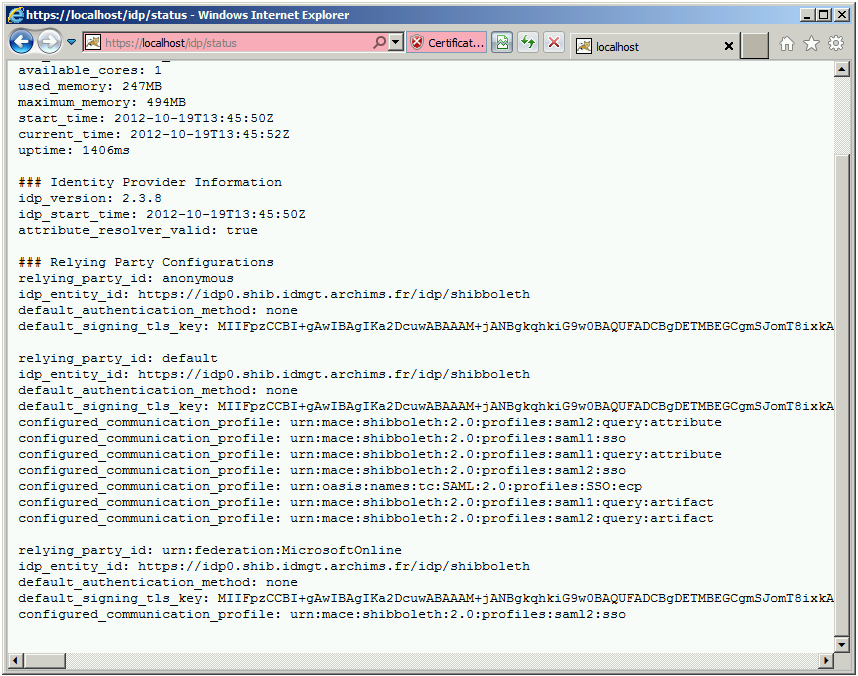
<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>

</AttributeAuthorityDescriptor>

</EntityDescriptor>

The first role descriptor *<IDPSSODescriptor>* element corresponds to the identity provider role of the Shibboleth 2 IdP in which we are interested. As previously stated, you can see in the associated *protocolSupportEnumeration* parameter that the Shibboleth 2 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.

You retrieve this information in the Shibboleth 2 IdP status page:



Furthermore, modifications made to the *attribute-resolver.xml* and *attribute-filter.xml* files enable the Shibboleth 2 IdP to issue to the Windows Azure AD relying party SAML 2.0 assertions, which contains claims sourced from the on-premise LDAP AD LDS data source.

These claims allow the Windows Azure AD sign-in service to match the user to a shadow identity 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 Windows Azure AD in accordance of one of the *<NameIDFormat>* elements in the Windows Azure AD metadata.

This corresponds to the unique identifier of the user tied to provisioning of the user in the Windows Azure AD tenant of the organization, i.e. the Windows 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 § 4.6 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 Windows Azure AD.

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

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

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

ID="\_bbe9366ec1870c87b48dd975086d4574"

IssueInstant="2012-10-22T11:07:13.169Z"

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.archims.fr/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.archims.fr/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="2012-10-22T11:12:13.169Z"

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

</saml2:SubjectConfirmation>

</saml2:Subject>

<saml2:Conditions NotBefore="2012-10-22T11:07:13.169Z"

NotOnOrAfter="2012-10-22T11:12:13.169Z">

<saml2:AudienceRestriction>

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

</saml2:AudienceRestriction>

</saml2:Conditions>

<saml2:AuthnStatement AuthnInstant="2012-10-22T11:07:13.091Z"

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.archims.fr

</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.archims.fr

</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.archims.fr/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-lived bearer token, i.e. without a proof of possession, that is issued by the Shibboleth 2 IdP to the Windows Azure AD sign-in service.

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 2 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)[[124]](#footnote-124) (SAMLCore) core specification of the OASIS SAML 2.0 standard.

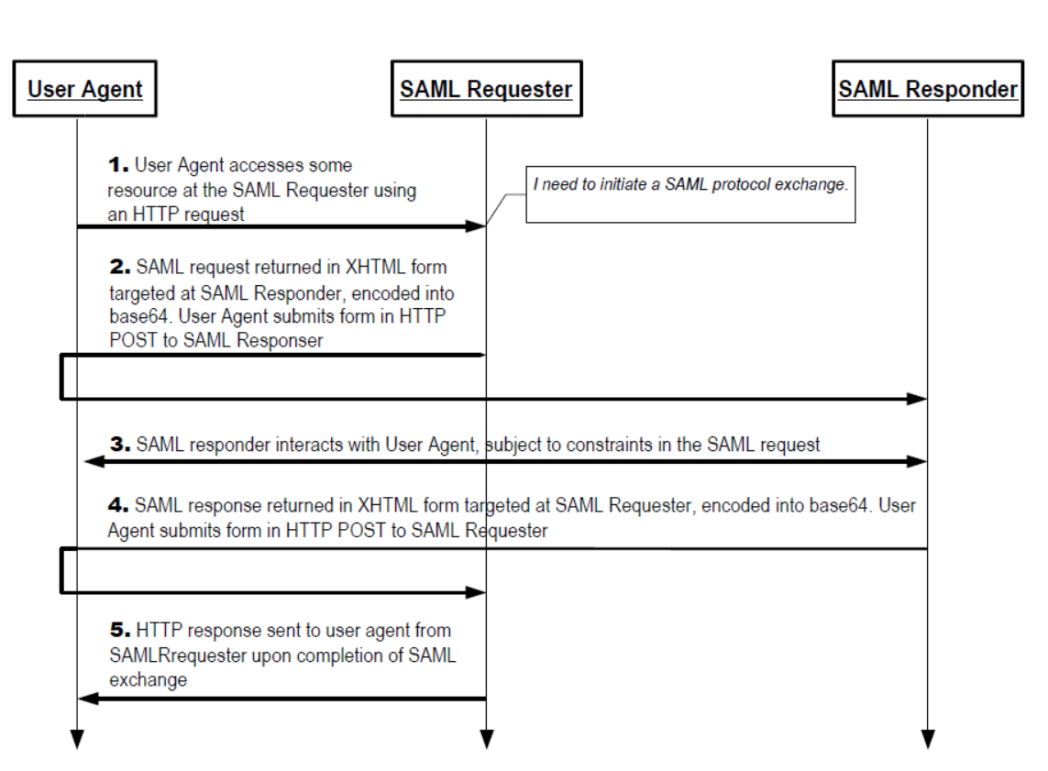
Finally, one should note that the Issuer URI, for example *https://idp0.shib.idmgt.archims.fr/idp/shibboleth*, in the above assertion is used to locate the namespace in the Windows Azure AD sign-in service.

It’s time to consider the two related authentication flows.

## Understanding the 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)[[125]](#footnote-125) (SAMLProf) document.

In the schema below, the SAML Requester is the Windows Azure AD sign-in service and the SAML Responder the Shibboleth 2 IdP. The User Agent is a browser.



This profile supports 12 possible deployment models (see section § 2.3 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)[[126]](#footnote-126) (SAMLBind) document.

It describes an SP-initiated SSO exchange where the HTTP Redirect Binding is used to deliver the *<AuthnRequest>* message to the Shibboleth 2 IdP.

The *<AuthnRequest>* message is the request generated by Windows Azure AD sign-in service for the federated domain to initiate the authentication request, which is through HTTPS XHTML form post to the Shibboleth 2 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="2012-10-22T11:07:02Z"

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 the Windows Azure AD.

The *<Response>* message is posted back by the Shibboleth 2 IdP to Windows Azure AD sign-in service’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="2012-10-22T11:07:13.169Z"

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.archims.fr/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="2012-10-22T11:07:13.169Z"

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

…

</saml2:Assertion>

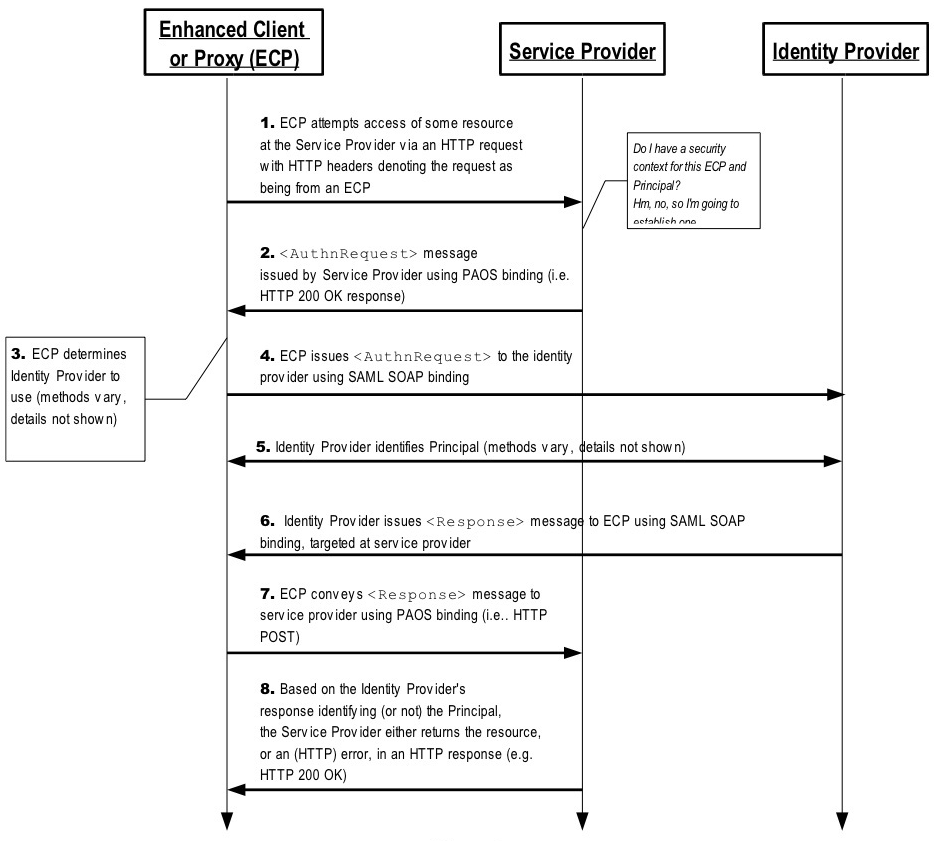
</saml2p:Response>

The creation of a cookie confirms a successful login.

## Understanding the ECP – Proxy Auth profile authentication flow

The SAML 2.0 Enhanced Client or Proxy (ECP) profile of SAML 2.0 is an adaptation of the above SAML profile used for Browser SSO with the parts that were designed around the limitations of a browser removed. In other words, in the ECP profile, the user agent is assumed to be something more than a browser (or perhaps a browser with a plugin for example).

The following diagram (extracted from 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)[[127]](#footnote-127) (SAMLProf) document) illustrates the authentication flow and related steps.



Please note that there is no browser redirect in this scenario. The service provider (SP), i.e. Windows Azure AD in our case, simply responds to the initial request from the client with a SAML 2.0 authentication request (in step 2).

The client can then make its own mind up about which IdP to submit the authentication request to. Please also note that the request might not be signed by the SP, allowing the client<->SP and client<->IdP protocol transactions to be decoupled. It's worth noting that the client might not even actually be a client, that's the reason why 'proxy' appears in the name of the SAML 2.0 profile.

**Note:**

*For more information, see the online help topic* [*ECP*](https://wiki.shibboleth.net/confluence/display/SHIB2/ECP)*[[128]](#footnote-128) on the Shibboleth Community wiki.*

The deployment model of this profile leverages the HTTP-POST-SimpleSign binding.

**Note:**

*For more information on the HTTP-POST-SimpleSign binding, see the* [*eponym specification*](https://www.oasis-open.org/committees/download.php/25607/sstc-saml-binding-simplesign-cd-02.pdf)*[[129]](#footnote-129).*

1. Shibboleth 2 Glossary/Concepts

**Attribute**: a piece of information about a user. Each attribute has a unique ID and has zero of more values. Shibboleth 2 attributes are protocol-agnostic data structures.

**Attribute definition**: a plugin that creates a single attribute by transforming other attributes and state information. Shibboleth 2 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 2 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 2 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 2 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 2, the recommended format is a URL IdP: *https://<FQDN hostname>/idp/shibboleth, respectively* SP: *http://<FQDN hostname>/shibboleth*.

**Login handler**: a Shibboleth 2 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 Windows Azure AD. Shibboleth 2 uses 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 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.

**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 SAML2.0 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 2 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 Windows 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 standards based, open source software package 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 implements widely used federated identity standards, principally OASIS SAML, to provide a federated single sign-on (SSO) and attribute exchange framework. Shibboleth 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 is developed in an open and participatory environment, is freely available, and is released under the Apache Software License.

**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.

1. Shibboleth 2 configuration file samples

**login.config Sample**

/\*

This is the JAAS configuration file used by the Shibboleth IdP.

A JAAS configuration file is a grouping of LoginModules defined in the following manner:

<LoginModuleClass> <Flag> <ModuleOptions>;

LoginModuleClass - fully qualified class name of the LoginModule class

Flag - indicates whether the requirement level for the modules;

allowed values: required, requisite, sufficient, optional

ModuleOptions - a space delimited list of name="value" options

For complete documentation on the format of this file see:

http://java.sun.com/j2se/1.5.0/docs/api/javax/security/auth/login/Configuration.html

For LoginModules available within the Sun JVM see:

http://java.sun.com/j2se/1.5.0/docs/guide/security/jaas/tutorials/LoginConfigFile.html

Warning: Do NOT use Sun's JNDI LoginModule to authentication against an LDAP directory,

Use the LdapLoginModule that ships with Shibboleth and is demonstrated below.

Note, the application identifier MUST be ShibUserPassAuth

\*/

ShibUserPassAuth {

// Example LDAP authentication

// See: https://spaces.internet2.edu/display/SHIB2/IdPAuthUserPass

edu.vt.middleware.ldap.jaas.LdapLoginModule required

ldapUrl="ldap://localhost:389"

baseDn="CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR"

bindDn="CN=ShibSvc,CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR"

bindCredential="Password1"

ssl="false"

tls="false"

userFilter="cn={0}"

subtreeSearch="false";

// Example Kerberos authentication, requires Sun's JVM

// See: https://spaces.internet2.edu/display/SHIB2/IdPAuthUserPass

/\*

com.sun.security.auth.module.Krb5LoginModule required

useKeyTab="true"

keyTab="/path/to/idp/keytab/file";

\*/

};

**attribute-resolver.xml Sample**

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

<!--

This file is an EXAMPLE configuration file. While the configuration presented in this

example file is functional, it isn't very interesting. However, there are lots of example

attributes, encoders, and a couple example data connectors.

Not all attribute definitions, data connectors, or principal connectors are demonstrated.

Deployers should refer to the Shibboleth 2 documentation for a complete list of components

and their options.

-->

<resolver:AttributeResolver xmlns:resolver="urn:mace:shibboleth:2.0:resolver"

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

xmlns:pc="urn:mace:shibboleth:2.0:resolver:pc"

xmlns:ad="urn:mace:shibboleth:2.0:resolver:ad"

xmlns:dc="urn:mace:shibboleth:2.0:resolver:dc"

xmlns:enc="urn:mace:shibboleth:2.0:attribute:encoder"

xmlns:sec="urn:mace:shibboleth:2.0:security"

xsi:schemaLocation="urn:mace:shibboleth:2.0:resolver classpath:/schema/shibboleth-2.0-attribute-resolver.xsd

urn:mace:shibboleth:2.0:resolver:pc classpath:/schema/shibboleth-2.0-attribute-resolver-pc.xsd

urn:mace:shibboleth:2.0:resolver:ad classpath:/schema/shibboleth-2.0-attribute-resolver-ad.xsd

urn:mace:shibboleth:2.0:resolver:dc classpath:/schema/shibboleth-2.0-attribute-resolver-dc.xsd

urn:mace:shibboleth:2.0:attribute:encoder classpath:/schema/shibboleth-2.0-attribute-encoder.xsd

urn:mace:shibboleth:2.0:security classpath:/schema/shibboleth-2.0-security.xsd">

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

<!-- Attribute Definitions -->

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

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

<resolver:AttributeDefinition id="ImmutableID"

xsi:type="ad:Simple"

xmlns="urn:mace:shibboleth:2.0:resolver:ad"

sourceAttributeID="uid">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="SAML2StringNameID"

xmlns="urn:mace:shibboleth:2.0:attribute:encoder"

nameFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" />

</resolver:AttributeDefinition>

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

<resolver:AttributeDefinition id="UserId"

xsi:type="ad:Simple"

sourceAttributeID="mail">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="IDPEmail"

friendlyName="UserId" />

</resolver:AttributeDefinition>

<!-- Schema: Core schema attributes-->

<!--

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="uid"

sourceAttributeID="uid">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:uid" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:0.9.2342.19200300.100.1.1"

friendlyName="uid" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="email"

sourceAttributeID="mail">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:mail" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:0.9.2342.19200300.100.1.3"

friendlyName="mail" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="homePhone"

sourceAttributeID="homePhone">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:homePhone" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:0.9.2342.19200300.100.1.20"

friendlyName="homePhone" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="homePostalAddress"

sourceAttributeID="homePostalAddress">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:homePostalAddress" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:0.9.2342.19200300.100.1.39"

friendlyName="homePostalAddress" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="mobileNumber"

sourceAttributeID="mobile">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:mobile" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:0.9.2342.19200300.100.1.41"

friendlyName="mobile" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="pagerNumber"

sourceAttributeID="pager">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:pager" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:0.9.2342.19200300.100.1.42"

friendlyName="pager" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="commonName"

sourceAttributeID="cn">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:cn" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.3"

friendlyName="cn" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="surname"

sourceAttributeID="sn">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:sn" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.4"

friendlyName="sn" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="locality"

sourceAttributeID="l">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:l" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.7"

friendlyName="l" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="stateProvince"

sourceAttributeID="st">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:st" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.8"

friendlyName="st" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="street"

sourceAttributeID="street">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:street" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.9"

friendlyName="street" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="organizationName"

sourceAttributeID="o">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:o" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.10" friendlyName="o" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="organizationalUnit"

sourceAttributeID="ou">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:ou" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.11"

friendlyName="ou" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple" id="title" sourceAttributeID="title">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:title" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.12"

friendlyName="title" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="postalAddress"

sourceAttributeID="postalAddress">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:postalAddress" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.16"

friendlyName="postalAddress" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="postalCode"

sourceAttributeID="postalCode">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:postalCode" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.17"

friendlyName="postalCode" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="postOfficeBox"

sourceAttributeID="postOfficeBox">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:postOfficeBox" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.18"

friendlyName="postOfficeBox" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="telephoneNumber"

sourceAttributeID="telephoneNumber">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:telephoneNumber" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.20"

friendlyName="telephoneNumber" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="givenName"

sourceAttributeID="givenName">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:givenName" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.42" friendlyName="givenName" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="initials"

sourceAttributeID="initials">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:initials" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.5.4.43"

friendlyName="initials" />

</resolver:AttributeDefinition>

-->

<!-- Schema: inetOrgPerson attributes-->

<!--

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="departmentNumber"

sourceAttributeID="departmentNumber">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:departmentNumber" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.16.840.1.113730.3.1.2"

friendlyName="departmentNumber" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="displayName"

sourceAttributeID="displayName">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:displayName" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.16.840.1.113730.3.1.241"

friendlyName="displayName" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="employeeNumber"

sourceAttributeID="employeeNumber">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:employeeNumber" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.16.840.1.113730.3.1.3"

friendlyName="employeeNumber" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="employeeType"

sourceAttributeID="employeeType">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:employeeType" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.16.840.1.113730.3.1.4"

friendlyName="employeeType" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="jpegPhoto"

sourceAttributeID="jpegPhoto">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:jpegPhoto" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:0.9.2342.19200300.100.1.60"

friendlyName="jpegPhoto" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="preferredLanguage"

sourceAttributeID="preferredLanguage">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:preferredLanguage" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:2.16.840.1.113730.3.1.39"

friendlyName="preferredLanguage" />

</resolver:AttributeDefinition>

-->

<!-- Schema: eduPerson attributes -->

<!-- We Generate the three of the four attributes required by the UK Federation

EpSA - statically defined as "member" (@scope)

EpPN - from LDAP sAMAccountName (@scope)

EpTID - (both flavors) -->

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="eduPersonAffiliation"

sourceAttributeID="eduPersonAffiliation">

<resolver:Dependency ref="staticAttributes" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:eduPersonAffiliation" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"

friendlyName="eduPersonAffiliation" />

</resolver:AttributeDefinition>

<!--

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="eduPersonEntitlement"

sourceAttributeID="eduPersonEntitlement">

<resolver:Dependency ref="staticAttributes" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:eduPersonEntitlement" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"

friendlyName="eduPersonEntitlement" />

</resolver:AttributeDefinition>

-->

<!-- EpPN is filled in for SAMAccountName (@scope) -->

<resolver:AttributeDefinition xsi:type="ad:Scoped"

id="eduPersonPrincipalName"

scope="shib.idmgt.archims.fr"

sourceAttributeID="sAMAccountName">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1ScopedString"

name="urn:mace:dir:attribute-def:eduPersonPrincipalName" />

<resolver:AttributeEncoder xsi:type="enc:SAML2ScopedString"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"

friendlyName="eduPersonPrincipalName" />

</resolver:AttributeDefinition>

<!-- EpSA - statically defined as "member" (@scope) -->

<resolver:AttributeDefinition xsi:type="ad:Scoped"

id="eduPersonScopedAffiliation"

scope="shib.idmgt.archims.fr"

sourceAttributeID="eduPersonAffiliation">

<resolver:Dependency ref="staticAttributes" />

<resolver:AttributeEncoder xsi:type="enc:SAML1ScopedString"

name="urn:mace:dir:attribute-def:eduPersonScopedAffiliation" />

<resolver:AttributeEncoder xsi:type="enc:SAML2ScopedString"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9"

friendlyName="eduPersonScopedAffiliation" />

</resolver:AttributeDefinition>

<!--

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="eduPersonAssurance"

sourceAttributeID="eduPersonAssurance">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:eduPersonAssurance" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.11"

friendlyName="eduPersonAssurance" />

</resolver:AttributeDefinition>

-->

<!-- EpTID (old) -->

<resolver:AttributeDefinition id="eduPersonTargetedID.old"

xsi:type="Scoped" xmlns="urn:mace:shibboleth:2.0:resolver:ad"

scope="shib.idmgt.archims.fr"

sourceAttributeID="computedID">

<resolver:Dependency ref="computedID" />

<resolver:AttributeEncoder xsi:type="SAML1ScopedString"

xmlns="urn:mace:shibboleth:2.0:attribute:encoder"

name="urn:mace:dir:attribute-def:eduPersonTargetedID" />

</resolver:AttributeDefinition>

<!-- EpTid (new) -->

<resolver:AttributeDefinition id="eduPersonTargetedID"

xsi:type="SAML2NameID"

xmlns="urn:mace:shibboleth:2.0:resolver:ad"

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

sourceAttributeID="computedID">

<resolver:Dependency ref="computedID" />

<resolver:AttributeEncoder xsi:type="SAML1XMLObject"

xmlns="urn:mace:shibboleth:2.0:attribute:encoder"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" />

<resolver:AttributeEncoder xsi:type="SAML2XMLObject"

xmlns="urn:mace:shibboleth:2.0:attribute:encoder"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"

friendlyName="eduPersonTargetedID" />

</resolver:AttributeDefinition>

<!-- Other EduPerson Attributes -->

<!--

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="eduPersonNickname"

sourceAttributeID="eduPersonNickname">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:eduPersonNickname" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.2"

friendlyName="eduPersonNickname" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="eduPersonOrgDN"

sourceAttributeID="eduPersonOrgDN">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:eduPersonOrgDN" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.3"

friendlyName="eduPersonOrgDN" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="eduPersonOrgUnitDN"

sourceAttributeID="eduPersonOrgUnitDN">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:eduPersonOrgUnitDN" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.4"

friendlyName="eduPersonOrgUnitDN" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="eduPersonPrimaryAffiliation"

sourceAttributeID="eduPersonPrimaryAffiliation">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:eduPersonPrimaryAffiliation" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.5"

friendlyName="eduPersonPrimaryAffiliation" />

</resolver:AttributeDefinition>

<resolver:AttributeDefinition xsi:type="ad:Simple"

id="eduPersonPrimaryOrgUnitDN"

sourceAttributeID="eduPersonPrimaryOrgUnitDN">

<resolver:Dependency ref="myLDAP" />

<resolver:AttributeEncoder xsi:type="enc:SAML1String"

name="urn:mace:dir:attribute-def:eduPersonPrimaryOrgUnitDN" />

<resolver:AttributeEncoder xsi:type="enc:SAML2String"

name="urn:oid:1.3.6.1.4.1.5923.1.1.1.8"

friendlyName="eduPersonPrimaryOrgUnitDN" />

</resolver:AttributeDefinition>

-->

<!-- Name Identifier related attributes -->

<resolver:AttributeDefinition id="transientId"

xsi:type="TransientId"

xmlns="urn:mace:shibboleth:2.0:resolver:ad">

<resolver:AttributeEncoder xsi:type="SAML1StringNameIdentifier"

xmlns="urn:mace:shibboleth:2.0:attribute:encoder"

nameFormat="urn:mace:shibboleth:1.0:nameIdentifier" />

<resolver:AttributeEncoder xsi:type="SAML2StringNameID"

xmlns="urn:mace:shibboleth:2.0:attribute:encoder"

nameFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:transient" />

</resolver:AttributeDefinition>

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

<!-- Data Connectors -->

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

<!-- Our Static Connector -->

<resolver:DataConnector id="staticAttributes"

xsi:type="Static"

xmlns="urn:mace:shibboleth:2.0:resolver:dc">

<Attribute id="eduPersonAffiliation">

<Value>member</Value>

</Attribute>

<Attribute id="eduPersonEntitlement">

<Value>urn:example.org:entitlement:entitlement1</Value>

<Value>urn:mace:dir:entitlement:common-lib-terms</Value>

</Attribute>

</resolver:DataConnector>

<!-- Example Relational Database Connector -->

<!--

<resolver:DataConnector id="mySIS"

xsi:type="RelationalDatabase"

xmlns="urn:mace:shibboleth:2.0:resolver:dc">

<ApplicationManagedConnection jdbcDriver="oracle.jdbc.driver.OracleDriver"

jdbcURL="jdbc:oracle:thin:@db.example.org:1521:SomeDB"

jdbcUserName="myid" jdbcPassword="mypassword" />

<QueryTemplate>

<![CDATA[

SELECT \* FROM student WHERE gzbtpid = '$requestContext.principalName'

]]>

</QueryTemplate>

<Column columnName="gzbtpid" attributeID="uid" />

<Column columnName="fqlft" attributeID="gpa" type="Float" />

</resolver:DataConnector>

-->

<!-- LDAP Connector - just attach to the AD LDAP -->

<resolver:DataConnector id="myLDAP"

xsi:type="LDAPDirectory"

xmlns="urn:mace:shibboleth:2.0:resolver:dc"

useStartTLS="false"

ldapURL="ldap://localhost:389"

baseDN="CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR"

principal="CN=ShibSvc,CN=Users,DC=SHIB,DC=IDMGT,DC=ARCHIMS,DC=FR"

principalCredential="Password1">

<FilterTemplate>

<![CDATA[

(cn=$requestContext.principalName)

]]>

</FilterTemplate>

<!-- We rely on the uniqueness of the objectSid. But it is binary so we \*must\* make it so -->

<!-- <LDAPProperty name="java.naming.ldap.attributes.binary" value="objectSid"/> -->

</resolver:DataConnector>

<!-- Computed targeted ID connector -->

<resolver:DataConnector xsi:type="ComputedId"

xmlns="urn:mace:shibboleth:2.0:resolver:dc"

id="computedID"

generatedAttributeID="computedID"

sourceAttributeID="objectSid"

salt="{9EAB897C-E2C0-4279-851A-38E5C1D4B848}">

<resolver:Dependency ref="myLDAP" />

</resolver:DataConnector>

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

<!-- Principal Connectors -->

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

<resolver:PrincipalConnector xsi:type="Transient"

xmlns="urn:mace:shibboleth:2.0:resolver:pc"

id="shibTransient"

nameIDFormat="urn:mace:shibboleth:1.0:nameIdentifier" />

<resolver:PrincipalConnector xsi:type="Transient"

xmlns="urn:mace:shibboleth:2.0:resolver:pc"

id="saml1Unspec"

nameIDFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified" />

<resolver:PrincipalConnector xsi:type="Transient"

xmlns="urn:mace:shibboleth:2.0:resolver:pc"

id="saml2Transient"

nameIDFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:transient" />

</resolver:AttributeResolver>

**attribute-filter.xml Sample**

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

<!--

This file is an EXAMPLE policy file. While the policy presented in this

example file is functional, it isn't very interesting.

Deployers should refer to the Shibboleth 2 documentation for a complete list of components

and their options.

-->

<afp:AttributeFilterPolicyGroup id="ShibbolethFilterPolicy"

xmlns:afp="urn:mace:shibboleth:2.0:afp"

xmlns:basic="urn:mace:shibboleth:2.0:afp:mf:basic"

xmlns:saml="urn:mace:shibboleth:2.0:afp:mf:saml"

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

xsi:schemaLocation="urn:mace:shibboleth:2.0:afp classpath:/schema/shibboleth-2.0-afp.xsd

urn:mace:shibboleth:2.0:afp:mf:basic classpath:/schema/shibboleth-2.0-afp-mf-basic.xsd

urn:mace:shibboleth:2.0:afp:mf:saml classpath:/schema/shibboleth-2.0-afp-mf-saml.xsd">

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

<afp:AttributeFilterPolicy id="UserId">

<afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString"

value="urn:federation:MicrosoftOnline" />

<afp:AttributeRule attributeID="UserId">

<afp:PermitValueRule xsi:type="basic:ANY"/>

</afp:AttributeRule>

</afp:AttributeFilterPolicy>

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

<afp:AttributeFilterPolicy id="ImmutableID">

<afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString"

value="urn:federation:MicrosoftOnline" />

<afp:AttributeRule attributeID="ImmutableID">

<afp:PermitValueRule xsi:type="basic:ANY"/>

</afp:AttributeRule>

</afp:AttributeFilterPolicy>

<!-- Release the transient ID, espa & eptid to anyone -->

<afp:AttributeFilterPolicy id="releaseToAnyone">

<afp:PolicyRequirementRule xsi:type="basic:ANY"/>

<!-- Transient -->

<afp:AttributeRule attributeID="transientId">

<afp:PermitValueRule xsi:type="basic:ANY"/>

</afp:AttributeRule>

<!-- EpSA -->

<afp:AttributeRule attributeID="eduPersonScopedAffiliation">

<afp:PermitValueRule xsi:type="basic:ANY" />

</afp:AttributeRule>

<!-- EpTid (both -->

<afp:AttributeRule attributeID="eduPersonTargetedID.old">

<afp:PermitValueRule xsi:type="basic:ANY" />

</afp:AttributeRule>

<afp:AttributeRule attributeID="eduPersonTargetedID">

<afp:PermitValueRule xsi:type="basic:ANY" />

</afp:AttributeRule>

</afp:AttributeFilterPolicy>

<!--

Release eduPersonEntitlement and the permissible values of eduPersonAffiliation

to three specific SPs

-->

<!--

<afp:AttributeFilterPolicy>

<afp:PolicyRequirementRule xsi:type="basic:OR">

<basic:Rule xsi:type="basic:AttributeRequesterString"

value="urn:example.org:sp:Portal" />

<basic:Rule xsi:type="basic:AttributeRequesterString"

value="urn:example.org:sp:SIS" />

<basic:Rule xsi:type="basic:AttributeRequesterString"

value="urn:example.org:sp:LMS" />

</afp:PolicyRequirementRule>

<afp:AttributeRule attributeID="eduPersonAffiliation">

<afp:PermitValueRule xsi:type="basic:OR">

<basic:Rule xsi:type="basic:AttributeValueString"

value="faculty"

ignoreCase="true" />

<basic:Rule xsi:type="basic:AttributeValueString"

value="student"

ignoreCase="true" />

<basic:Rule xsi:type="basic:AttributeValueString"

value="staff"

ignoreCase="true" />

<basic:Rule xsi:type="basic:AttributeValueString"

value="alum"

ignoreCase="true" />

<basic:Rule xsi:type="basic:AttributeValueString"

value="member"

ignoreCase="true" />

<basic:Rule xsi:type="basic:AttributeValueString"

value="affiliate"

ignoreCase="true" />

<basic:Rule xsi:type="basic:AttributeValueString"

value="employee"

ignoreCase="true" />

<basic:Rule xsi:type="basic:AttributeValueString"

value="library-walk-in"

ignoreCase="true" />

</afp:PermitValueRule>

</afp:AttributeRule>

</afp:AttributeFilterPolicy>

-->

<!--

Release the given name of the user to our portal service provider

-->

<!--

<afp:AttributeFilterPolicy>

<afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString"

value="urn:example.org:sp:myPortal" />

<afp:AttributeRule attributeID="givenName">

<afp:PermitValueRule xsi:type="basic:ANY" />

</afp:AttributeRule>

</afp:AttributeFilterPolicy>

-->

</afp:AttributeFilterPolicyGroup>

1. Microsoft Office 365: http://office365.microsoft.com/ [↑](#footnote-ref-1)
2. Office 365 Help: http://onlinehelp.microsoft.com/en-us/office365-enterprises/ [↑](#footnote-ref-2)
3. Office 365 Deployment Guide for Enterprises: http://www.microsoft.com/download/en/details.aspx?id=26509 [↑](#footnote-ref-3)
4. Office 365 Tech Center web site: http://technet.microsoft.com/en-us/office365/default [↑](#footnote-ref-4)
5. Office 365 Community web site: http://community.office365.com/en-us/default.aspx [↑](#footnote-ref-5)
6. Windows Azure Active Directory: http://technet.microsoft.com/en-us/library/hh967619.aspx [↑](#footnote-ref-6)
7. Windows Azure Active Directory Sign-up: http://g.microsoftonline.com/0AX00en/5 [↑](#footnote-ref-7)
8. Preview of the Windows Azure AD Management Portal: https://activedirectory.windowsazure.com/ [↑](#footnote-ref-8)
9. Windows Intune: http://technet.microsoft.com/en-us/windows/intune.aspx [↑](#footnote-ref-9)
10. Active Directory Federation Services 2.0 RTW: http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=10909 [↑](#footnote-ref-10)
11. Article 2607496 Description of Update Rollup 1 for Active Directory Federation Services (AD FS) 2.0: http://support.microsoft.com/kb/2607496 [↑](#footnote-ref-11)
12. Article 2681584 Description of Update Rollup 2 for Active Directory Federation Services (AD FS) 2.0: http://support.microsoft.com/kb/2681584 [↑](#footnote-ref-12)
13. Microsoft Office 365 Single Sign-On (SSO) with AD FS 2.0: http://www.microsoft.com/en-us/download/details.aspx?id=28971 [↑](#footnote-ref-13)
14. Microsoft Office 365: Identity   
    and Access Solutions: http://channel9.msdn.com/Events/TechEd/NorthAmerica/2011/OSP215 [↑](#footnote-ref-14)
15. Optimal IDM Virtual Identity Server Federation Services: http://go.microsoft.com/fwlink/?LinkID=266318 [↑](#footnote-ref-15)
16. PingFederate 6.10: http://go.microsoft.com/fwlink/?LinkID=266320 [↑](#footnote-ref-16)
17. Security Assertion Markup Language (SAML) 2.0: http://go.microsoft.com/fwlink/?LinkId=193996 [↑](#footnote-ref-17)
18. Identity federation definition from Wikipedia: http://en.wikipedia.org/wiki/Federated\_identity [↑](#footnote-ref-18)
19. Install and Upgrade the Microsoft Online Services Directory Synchronization tool: http://onlinehelp.microsoft.com/en-us/office365-enterprises/ff652545.aspx [↑](#footnote-ref-19)
20. Configure directory synchronization: http://technet.microsoft.com/en-us/library/hh967629.aspx [↑](#footnote-ref-20)
21. Use third-party identity providers to implement single sign-on: http://technet.microsoft.com/en-us/library/jj679342.aspx [↑](#footnote-ref-21)
22. Web SSO to Office 365 Using PingFederate: http://go.microsoft.com/fwlink/?LinkID=266321 [↑](#footnote-ref-22)
23. Shibboleth 2 home page: http://go.microsoft.com/fwlink/?LinkID=256497 [↑](#footnote-ref-23)
24. Internet2/MACE (Middleware Architecture Committee for Education): http://middleware.internet2.edu/MACE [↑](#footnote-ref-24)
25. Shibboleth Consortium: http://shibboleth.net [↑](#footnote-ref-25)
26. OASIS Security Services (SAML) TC: http://www.oasis-open.org/committees/tc\_home.php?wg\_abbrev=security [↑](#footnote-ref-26)
27. Now replaced by the discovery service (see later in this chapter). [↑](#footnote-ref-27)
28. Shibboleth 2.0: http://go.microsoft.com/fwlink/?LinkID=256497 [↑](#footnote-ref-28)
29. Liberty Identity Federation Framework (ID-FF) 1.2 : http://www.projectliberty.org/resource\_center/specifications/liberty\_alliance\_id\_ff\_1\_2\_specifications/?f=resource\_center/specifications/liberty\_alliance\_id\_ff\_1\_2\_specifications [↑](#footnote-ref-29)
30. Liberty Interoperable certification program: http://www.projectliberty.org/liberty/liberty\_interoperable [↑](#footnote-ref-30)
31. Drummond group Web site: http://www.drummondgroup.com/ [↑](#footnote-ref-31)
32. Kantara initiative: http://kantarainitiative.org/ [↑](#footnote-ref-32)
33. SAML V2.0 Executive Overview: http://www.oasis-open.org/committees/download.php/13525/sstc-saml-exec-overview-2.0-cd-01-2col.pdf [↑](#footnote-ref-33)
34. SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0 TECHNICAL OVERVIEW: http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf [↑](#footnote-ref-34)
35. 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-35)
36. 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-36)
37. 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-37)
38. 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-38)
39. Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf [↑](#footnote-ref-39)
40. Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf [↑](#footnote-ref-40)
41. Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf [↑](#footnote-ref-41)
42. Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf [↑](#footnote-ref-42)
43. OASIS eXtensible Access Control Markup Language (XACML) TC: http://www.oasis-open.org/committees/tc\_home.php?wg\_abbrev=xacml [↑](#footnote-ref-43)
44. Shibboleth Service Provider: http://shibboleth.net/products/service-provider.html [↑](#footnote-ref-44)
45. Shibboleth Identity Provider: http://shibboleth.net/products/identity-provider.html [↑](#footnote-ref-45)
46. Understanding Shibboleth: https://wiki.shibboleth.net/confluence/display/SHIB2/UnderstandingShibboleth [↑](#footnote-ref-46)
47. Shibboleth Architecture Technical Overview: http://open-systems.ufl.edu/files/draft-mace-shibboleth-tech-overview-latest.pdf [↑](#footnote-ref-47)
48. federation Education-Recherche web site: https://services.renater.fr/federation/index [↑](#footnote-ref-48)
49. Identity Provider Discovery Service Protocol and Profile specification: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf [↑](#footnote-ref-49)
50. You must already have an installed and configured Shibboleth SP, version 2.4 or above in order to use the EDS [↑](#footnote-ref-50)
51. Index of /downloads/service-provider/latest: http://shibboleth.net/downloads/service-provider/latest/ [↑](#footnote-ref-51)
52. Service CAS (Client Authentication Service): http://www.jasig.org/cas [↑](#footnote-ref-52)
53. CAS became a Jasig project in December 2004. [↑](#footnote-ref-53)
54. Shibboleth-CAS Integration: https://wiki.jasig.org/display/CASUM/Shibboleth-CAS+Integration [↑](#footnote-ref-54)
55. Index of /downloads/identity-provider/latest: http://shibboleth.net/downloads/identity-provider/latest/ [↑](#footnote-ref-55)
56. Security Assertion Markup Language (SAML) 2.0: http://go.microsoft.com/fwlink/?LinkId=193996 [↑](#footnote-ref-56)
57. RFC 2141 URN syntax: http://tools.ietf.org/html/rfc2141 [↑](#footnote-ref-57)
58. NativeSPMetadataProvider: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPMetadataProvider [↑](#footnote-ref-58)
59. Office 365 Identity Service Description: http://www.microsoft.com/download/en/details.aspx?id=13602 [↑](#footnote-ref-59)
60. User-Principal-Name attribute: http://msdn.microsoft.com/en-us/library/windows/desktop/ms680857(v=vs.85).aspx [↑](#footnote-ref-60)
61. RFC 822 Standard for ARPA Internet Text Messages: http://tools.ietf.org/html/rfc822 [↑](#footnote-ref-61)
62. Single sign-on roadmap: http://technet.microsoft.com/en-us/library/hh967643 [↑](#footnote-ref-62)
63. Use Shibboleth Identity Provider to implement single sign-on: http://technet.microsoft.com/en-us/library/jj205456 [↑](#footnote-ref-63)
64. Prepare for single sign-on: http://technet.microsoft.com/en-us/library/jj151786 [↑](#footnote-ref-64)
65. Microsoft Office 365 for enterprises Deployment Readiness Tool: http://g.microsoftonline.com/0BD00en-US/506 [↑](#footnote-ref-65)
66. Microsoft Office 365 Deployment Guide: http://community.office365.com/en-us/f/183/p/1541/5095.aspx#5095 [↑](#footnote-ref-66)
67. ADModify: http://admodify.codeplex.com/ [↑](#footnote-ref-67)
68. Management Agents in FIM 2010: http://technet.microsoft.com/en-us/library/ff608275(WS.10).aspx [↑](#footnote-ref-68)
69. OpenLDAP Extensible Management Agent (XMA) connector for FIM 2010: http://sourceforge.net/projects/openldap-xma/ [↑](#footnote-ref-69)
70. AD FS 2.0 Step-by-Step Guide: Federation with Shibboleth 2 and the InCommon Federation: http://go.microsoft.com/fwlink/?LinkId=204784 [↑](#footnote-ref-70)
71. AD FS 2.0 Step-by-Step and How To guides: http://technet.microsoft.com/en-us/library/dd727938(WS.10).aspx [↑](#footnote-ref-71)
72. Windows Server 2008 R2 Evaluation Virtual Hard Drive Images for Hyper-V (180 Days): http://go.microsoft.com/fwlink/?LinkId=179736. [↑](#footnote-ref-72)
73. Shibboleth Installation: https://wiki.shibboleth.net/confluence/display/SHIB2/Installation [↑](#footnote-ref-73)
74. [Configuration](https://wiki.shibboleth.net/confluence/display/SHIB2/Configuration): https://wiki.shibboleth.net/confluence/display/SHIB2/Configuration [↑](#footnote-ref-74)
75. 816042 How to configure an authoritative time server in Windows Server: http://support.microsoft.com/kb/816042 [↑](#footnote-ref-75)
76. Jetty Java Web Server: http://jetty.mortbay.org/jetty [↑](#footnote-ref-76)
77. Oracle JRE 7u7: http://go.microsoft.com/fwlink/?LinkId=204015 [↑](#footnote-ref-77)
78. Convert a PFX to JKS using Windows: http://www.agrypnia.com/blog/2009/07/28/convert-a-pfx-to-jks-using-windows.html [↑](#footnote-ref-78)
79. Private Key PFX to/from JKS Conversion Using OpenSSL and Jetty: http://andrius.miasnikovas.lt/2010/10/private-key-pfx-jks-conversion-using-openssl-and-jetty/ [↑](#footnote-ref-79)
80. OpenLDAP Software: http://www.openldap.org/ [↑](#footnote-ref-80)
81. Active Directory Lightweight Directory Services: http://technet.microsoft.com/en-us/library/cc731868(WS.10).aspx [↑](#footnote-ref-81)
82. Step 1: Install the AD LDS Server Role: http://technet.microsoft.com/en-us/library/cc754486(v=ws.10).aspx [↑](#footnote-ref-82)
83. Shibboleth IdP download site: http://shibboleth.net/downloads/identity-provider/latest/ [↑](#footnote-ref-83)
84. Apache Tomcat 6.x : http://tomcat.apache.org/download-60.cgi [↑](#footnote-ref-84)
85. IdP ECP Extension: https://wiki.shibboleth.net/confluence/display/SHIB2/IdP+ECP+Extension [↑](#footnote-ref-85)
86. *IdPConfiguration*: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPConfiguration [↑](#footnote-ref-86)
87. IdPAuthUserPass: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAuthUserPass [↑](#footnote-ref-87)
88. JAAS Tutorials: http://docs.oracle.com/javase/1.5.0/docs/guide/security/jaas/tutorials/index.html [↑](#footnote-ref-88)
89. IdPAuthUserPass: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAuthUserPass [↑](#footnote-ref-89)
90. [ResolverLDAPDataConnector](https://wiki.shibboleth.net/confluence/display/SHIB2/ResolverLDAPDataConnector): https://wiki.shibboleth.net/confluence/display/SHIB2/ResolverLDAPDataConnector [↑](#footnote-ref-90)
91. IdPEnableECP: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPEnableECP [↑](#footnote-ref-91)
92. IdPLogging: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPLogging [↑](#footnote-ref-92)
93. Shibboleth troubleshooting: https://wiki.shibboleth.net/confluence/display/SHIB2/Troubleshooting [↑](#footnote-ref-93)
94. Shibboleth troubleshooting: https://wiki.shibboleth.net/confluence/display/SHIB2/Troubleshooting [↑](#footnote-ref-94)
95. IdPStatus: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPStatus [↑](#footnote-ref-95)
96. Configure Shibboleth for use with single sign-on: http://technet.microsoft.com/en-us/library/jj205463 [↑](#footnote-ref-96)
97. *IdPRelyingParty*: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPRelyingParty [↑](#footnote-ref-97)
98. *IdPMetadataProvider*: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPMetadataProvider [↑](#footnote-ref-98)
99. *IdPAddAttribute*: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAddAttribute [↑](#footnote-ref-99)
100. *IdPAddAttributeFilter*: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAddAttributeFilter [↑](#footnote-ref-100)
101. Shibboleth troubleshooting: https://wiki.shibboleth.net/confluence/display/SHIB2/Troubleshooting [↑](#footnote-ref-101)
102. Install Windows PowerShell for single sign-on with Shibboleth: http://technet.microsoft.com/en-us/library/jj205464 [↑](#footnote-ref-102)
103. Windows PowerShell Web site: http://www.microsoft.com/powershell [↑](#footnote-ref-103)
104. Windows PowerShell online help: http://technet.microsoft.com/en-us/library/bb978526.aspx [↑](#footnote-ref-104)
105. Windows PowerShell Weblog: http://blogs.msdn.com/powershell [↑](#footnote-ref-105)
106. Windows PowerShell SDK: http://msdn2.microsoft.com/en-us/library/aa830112.aspx [↑](#footnote-ref-106)
107. Description of Microsoft Online Services Sign-In Assistant (MOS SIA): http://community.office365.com/en-us/w/office/534.aspx [↑](#footnote-ref-107)
108. Microsoft Online Services Sign-In Assistant for IT Professionals RTW: http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=28177 [↑](#footnote-ref-108)
109. Set up your desktop for Office 365: http://onlinehelp.microsoft.com/en-us/Office365-enterprises/ff637594.aspx [↑](#footnote-ref-109)
110. Microsoft Online Services Module for Windows PowerShell (64-bit version): http://go.microsoft.com/fwlink/p/?linkid=236293 [↑](#footnote-ref-110)
111. Use Windows PowerShell cmdlets to manage your Windows Azure AD tenant: http://technet.microsoft.com/en-us/library/jj151805 [↑](#footnote-ref-111)
112. Windows PowerShell cmdlet descriptions: http://technet.microsoft.com/en-us/library/jj151835.aspx [↑](#footnote-ref-112)
113. Set up a trust between Shibboleth and Windows Azure AD: http://technet.microsoft.com/en-us/library/jj205457 [↑](#footnote-ref-113)
114. Add your domain: http://technet.microsoft.com/en-us/library/hh969247.aspx [↑](#footnote-ref-114)
115. Verify a domain at any domain name registrar: http://technet.microsoft.com/en-us/library/jj151803.aspx [↑](#footnote-ref-115)
116. Configure directory synchronization: http://technet.microsoft.com/en-us/library/hh967629.aspx [↑](#footnote-ref-116)
117. Verify single sign-on with Shibboleth: http://technet.microsoft.com/en-us/library/jj205458 [↑](#footnote-ref-117)
118. Fiddler2: http://www.fiddler.com [↑](#footnote-ref-118)
119. SAML tracer: https://addons.mozilla.org/en-US/firefox/addon/saml-tracer/ [↑](#footnote-ref-119)
120. [IdPAuthUserPassLoginPage](https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAuthUserPassLoginPage): https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAuthUserPassLoginPage [↑](#footnote-ref-120)
121. Shibboleth troubleshooting: https://wiki.shibboleth.net/confluence/display/SHIB2/Troubleshooting [↑](#footnote-ref-121)
122. Troubleshoot single sign-on: http://technet.microsoft.com/en-us/library/jj151834.aspx [↑](#footnote-ref-122)
123. 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-123)
124. 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-124)
125. 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-125)
126. 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-126)
127. 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-127)
128. ECP: https://wiki.shibboleth.net/confluence/display/SHIB2/ECP [↑](#footnote-ref-128)
129. SAMLv2.0 HTTP POST “SimpleSign” Binding: https://www.oasis-open.org/committees/download.php/25607/sstc-saml-binding-simplesign-cd-02.pdf [↑](#footnote-ref-129)