WS-Federation IdP Operation

Contacts

Ilex International

51 boulevard Voltaire

92600 Asnières-sur-Seine

FRANCE

Telephone : +33 (0)1 46 88 03 40

Fax : +33 (0)1 46 88 03 41

http://www.ilex-international.com

support@ilex-international.com

Date: 08/07/14

Legal Information

Table of Contents

1. Overview of the WS-Federation architecture

In this section:

1.1 General

WS-Federation is an identity federation architecture which is implemented by Microsoft, in particular. This architecture is based on the **WS-Trust** suite of open protocols.

Like any identity federation architecture, WS-Federation defines “identity providers” along with “service providers”.

In the WS-Federation context, the identity provider is called “STS” (Security Token Service) and the service provider is called “Relying Party”.

Microsoft applications likely to delegate user authentication to the STSs of the federation comply with the **Windows Identity Foundation** specification, whose API is implemented in the **dotNet** Framework.

The WS-Federation architecture is particularly useful for application platforms running in **cloud** or **SaaS** mode, such as Microsoft Azure and Microsoft Office 365.

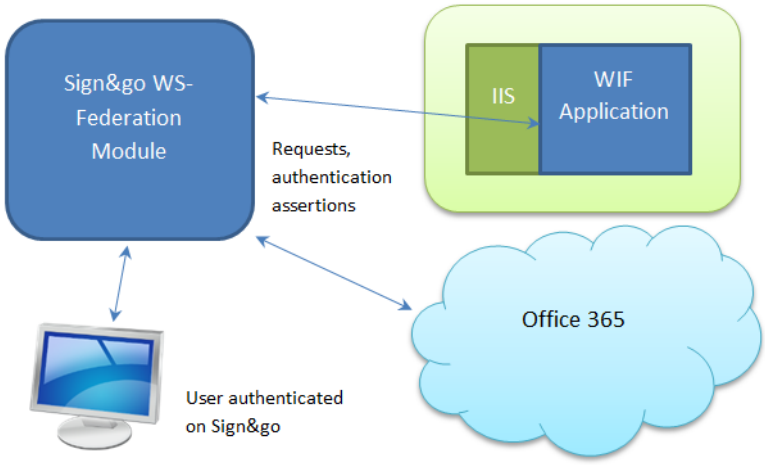

Illustration 1 • 1 : Sign&go architecture in WS-Federation

1.2 Sign&go in the WS-Federation architecture

In a WS-Federation architecture, Sign&go is the federation's STS.

In this configuration, the user authenticates to his Sign&go infrastructure, and the applications in the federation that have a trust relationship with Sign&go recognise this authentication.

This trust relationship relies on the exchange of signed WS-Trust messages.

Sign&go supports the following federation modes:

Federation mode	Description	Example of use
Passive	Authentication mode that the applications running in a Web browser use.	Authentication of "Microsoft SharePoint" users
Active	Authentication that "rich clients" use. ***Note :*** *As a reminder, "rich clients" are the applications directly based on a computer or a tablet, which run without a Web browser.*	Authentication of users of "Microsoft Outlook" or "Microsoft Lync" in the context of the "Microsoft Office 365" architecture

1.3 Entry points provided by Sign&go

The Sign&go authentication application provides its clients with the following entry points:

Entry point	Description
Distribution point of the federation metadata	**/WSFed/trust/mex**
Passive federation access point	**/WSFed/ls**
Active federation entry points	Entry points to SOAP type Web services. These entry points deliver WS-Trust security tokens in response to authentication requests. Two active federation entry points are implemented in Sign&go. These are mostly used for rich clients in Microsoft Office 365. These entry points are as follows: ■ Entry point used in response to authentication requests in "login/password" mode: **/WSFed/usernamemixed** ■ Entry point used in response to authentication requests in Kerberos mode: **/WSFed/windowstransport**

If you want, you can suffix the federation entry points with the Sign&go name of the IdP in question. For example, an IdP with the **MyWSFedIdp** name in the Sign&go configuration provides the following entry points:

- /WSFed/ls/**MyWSFedIdp**
- /WSfed/usernamemixed/**MyWSFedIdp**
- /WSFed/windowstransport/**MyWSFedIdp**

The specific **default** IdP name, provides the following non-suffixed access points:

- /WSFed/ls
- /WSFed/usernamemixed
- /WSFed/windowstransport

*In the case of a link with Microsoft Office 365, the configured IdP must have the **default** reserved name.*

2. Sign&go configuration

In this section:

2.1 General configuration

- To configure Sign&go so that it plays the part of the STS (identity provider), you must perform the following operations in the Sign&go administration console:
  - Define the URLs of the WS-Federation module access points and tune them with the URLs of the applications that trust the WS-Federation module.

    Indeed, the WS-Federation module issues messages towards URLs which you have to configure in the Sign&go administration console to let Sign&go know where to send these messages. Conversely, the federation's partner must in turn declare and configure URLs that Sign&go uses to receive messages.
  - Define the approved signature keys and certificates.

*For more information on the general configuration to perform in Sign&go, refer to the **Help** of the Sign&go administration console.*

2.2 Microsoft Office 365 configuration (special case)

The Sign&go configuration for Microsoft Office 365 is an important special case. For more information, refer to section Sign&go and Microsoft Office 365 (special case) (page 13).

3. Sign&go and Microsoft Office 365 (special case)

In this section:

3.1 Federation principles

Concerning the link between Sign&go and Microsoft Office 365, the usual identity federation themes apply, Microsoft Office 365 playing the part of the SP and Sign&go that of the IdP.

This section presents:

3.1.1 Link mode between Microsoft Office 365 accounts and Sign&go accounts

The federation key between the Microsoft Office 365 accounts and Sign&go is the `ImmutableID` attribute of the Windows user. This `ImmutableID` attribute which is transmitted to Microsoft Office 365 in the Sign&go assertion, allows Microsoft Office 365 to find the identity of the Office user concerned.

Reminder: *During registration of the user in the Active Directory directory, Active Directory assigns him (or her) the* `ImmutableID` *attribute which it stores in binary format in the* `objectGUID` *attribute of the user.*

The creation of a user on Microsoft Office 365 consists in running a create user command on Microsoft Office 365. This command serves to federate the local `ImmutableID` attribute of the Windows user with the Microsoft Office 365 account.

The creation command of a federated user on Microsoft Office 365 uses the user's local `ImmutableID` *attribute as parameter (see Create federated users on Microsoft Office 365, page 23).*

The `UserPrincipalName` *attribute is also used for linking a local account with Microsoft Office 365.*

- You must initialise the `UserPrincipalName` attribute with the e-mail address of the user on Microsoft Office 365.

3.1.2 Authentication mechanisms

Microsoft Office 365 provides services in light client mode as well as in rich client mode. Authentication modes vary according to the type of client:

- Authentication mechanism for light clients (page 14)
- Authentication mechanisms for rich clients (page 14)

3.1.2.1 Authentication mechanism for light clients

Basically, the authentication mechanism for light clients is as follows:

1. The user authenticates to Sign&go via his (or her) Web browser using any authentication scheme.
2. The user connects to the Microsoft Office 365 site.
3. As a result of the transfer of a Sign&go assertion to the Microsoft Office 365 relying party, the user is authenticated to the Microsoft Office 365 account which is linked to his Sign&go account.

3.1.2.2 Authentication mechanisms for rich clients

The rich client authentication mode varies depending on whether the user is registered in a Windows domain or not:

If the user...	...then
is **registered** in a Windows domain and that the controller of this domain is currently accessible	The rich client, for example Microsoft Outlook 2013, uses the Kerberos token of the current session on the local client workstation, to send a Web service request to the **/WSFed/windowstransport** access point of the Sign&go authentication application. The authentication process is then completely transparent for the user who is authenticated on his Windows session.
is **not registered** in a Windows domain or, the domain is currently not accessible (as in the case of a mobile user)	Microsoft Outlook 2013 addresses the **/WSFed/usernamemixed** access point using the user's login and password. Then, Microsoft Outlook 2013 may invite the user to specify his login and password, which he can save in Microsoft Outlook 2013 for subsequent connections.

The **Microsoft Online Sign in assistant** service takes charge of the federated authentication mechanisms of rich clients. This service is available on Microsoft sites and is automatically installed on the user's workstation when the user downloads his updated rich client version from the Microsoft Online site.

In terms of federation, you have no specific configuration to perform on the client workstation.

When you register the client domain with Microsoft Office 365, you must only declare the Sign&go access points to the Microsoft Online service (see Register the client domain with Microsoft Office 365, page 15).

3.2 Configuration of the connections to Microsoft Office 365

In order for Microsoft Office 365 to recognise the Sign&go STS as eligible for the authentication of the users of the client domain, you must:

1. Register the client domain with Microsoft Office 365 (page 15)
2. Configure the WS-Federation module and its associated Kerberos and login/password schemes (page 18)
3. Create federated users on Microsoft Office 365 (page 23)

3.2.1 Register the client domain with Microsoft Office 365

This section gives a general overview of the registration of the client domain with Microsoft Office 365, taking the airline.com site as an example. For more information on the complete registration and operation procedures of the client domain with Microsoft Office 365, refer to the Microsoft Office 365 Web site.

About extranet and intranet domain names:

In our configuration example, we refer to the general case where the DNS domain name of the Windows domain controller of the intranet is different than the Microsoft Office 365 extranet domain name:

Extranet domain name	airline.com
Intranet domain name	airline.lan

However, extranet and intranet domain names can be identical with no impact on operations.

- *To get this configuration case, in our configuration example, you must change all the **.lan** to **.com**.*

You should also ensure that you have a good understanding of these two different domain names to use them.

If the external domain name is identical to the internal one, you must pay particular attention to the common confusion problems related to domain name resolution, in the case where the user needs to access resources both on the extranet and on the intranet domains.

Guide in the various configuration choices:

*Rich clients send authentication requests to **WS-Federation** Web services on the basis of the external domain name (**airline.com** in our example). If an **airline.com** area is declared in the intranet and if the domain name resolutions on the extranet and intranet point to the same resources, there will be no problem.*

- *Therefore, you must perform the following tests:*
  - *Test all the use cases of the solution and check the following points:*
    - *The rich clients located in the domain must benefit from the Kerberos authentication which is "Windows integrated": connections of rich clients to Office are transparent, with no login/password entries.*
    - *Rich clients can authenticate in login/password mode when the domain controller is not accessible.*
  - *Test computers, mobiles and tablets for the various cases.*

To register a client domain with Microsoft Office 365, proceed as follows:

1. Register the airline.com client Web site with Microsoft Office 365:
   1. Connect to the Microsoft Office 365 Web site.
   2. Register the airline.com client Web site and declare a "myadmin" administrator.

      The Microsoft Office 365 Web site then generates the full name for administrator "myadmin", as follows: myadmin@airline.onmicrosoft.com.

To validate the registration of the airline.com client Web site, Microsoft Office 365 requests a "proof of ownership" of the Web site. One of the methods to provide Microsoft Office 365 with this proof of ownership is to add a TXT record in the domain's DNS. The "myadmin" administrator of the airline.com client Web site must temporarily add a label provided by Microsoft Office 365 in the domain's DNS.

   3. Declare the users of the Microsoft Office 365 service.

You must respect the constraints related to the creation of user accounts within Microsoft.

2. To manage the airline.com client Web site from the airline.com site itself:
   1. On the Windows machine that you have configured to administer the Microsoft Office 365 service, install the libraries of the **Windows Azure Active Directory Module for Windows PowerShell** from the Microsoft site.
   2. To manage the airline.com client Web site from the airline.com site itself, execute the PowerShell command desired on the Windows machine that administers the Microsoft Office 365 service.
3. Connect the Microsoft Office 365 platform to the WS-Federation IdP:
   1. Verify that you have an RSA signature certificate equipped with a 1024-bit key, and if you do not, make sure that you obtain one.

This certificate will be used to sign the assertions that the Sign&go IdP produces.

2. Configure the RSA signature certificate as indicated below, **in both places**:

To configure the RSA signature certificate...	...proceed as follows:
in Sign&go	▶ In the Sign&go administration console > **Partners** menu > then, in the left tree: **WS-Federation IdPs** > then at the centre of the page, section **Signature certificate**, configure the following parameters: ■ **Certificates keystore path** ■ **Keystore password** ■ **Alias of the signature certificate**

at the level of the Microsoft Office 365 account	▶ Run the PowerShell command **Set-MsolDomainAuthentication** below, with the appropriate parameters.
	Example of PowerShell command flow:
	`# connect to the Office 365 site as administrator`
	`# enter the administrator's name and password in the Get-credential dialogue box,`
	`# set the $cred variable:`
	`PS C:\>$cred=Get-credential`
	`# connect to the Office 365 site`
	`PS C:\>Connect-MsolService -Credential $cred`
	
	`# set the PowerShell variables for the Set-MsolDomainAuthentication command:`
	
	`PS C:\>$Dom = "airline.com"`
	`PS C:\>$acturi = "https://air.airline.com/WSFed/usernamemixed"`
	`PS C:\>$branN = "AIRLINE-COM"`
	`PS C:\>$Issuer = "https://air.airline.com/signandgo"`
	`PS C:\>$LogOff = "https://air.airline.com/auth/logout.jsp"`
	`PS C:\>$mex = "https://air.airline.com/WSFed/trust/mex"`
	`PS C:\>$nexcert = ""`
	`PS C:\>$passive = "https://air.airline.com/WSFed/ls/default"`
	`PS C:\>$cert ="`
	`MIIESDCCAzCgAwIB.....u3xWJiIVsiQWxCe0794/ICzdPw="`
	
	`# link to the Office 365 site, federate the domain`
	`Set-MsolDomainAuthentication –DomainName $Dom -FederationBrandName $branN -Authentication Federated -PassiveLogOnUri $passive -SigningCertificate $cert -IssuerUri $Issuer -ActiveLogOnUri $acturi -LogOffUri $LogOff`
	
	Note : *You must pass the above PowerShell command* ***Set-MsolDomainAuthentication*** *to the* ***airline.com*** *domain federation once only. For subsequent maintenance operations, you will have to use the PowerShell command* ***Set-MsolDomainFederationSettings****.*

3.2.2 Configure the WS-Federation module and its associated Kerberos and login/password schemes

Configuring the WS-Federation module for Microsoft Office 365 amounts to configuring the WS-Federation IdP as indicated in Table 1 • 1 - Example of configuration of the WS-Federation IdP for Microsoft Office 365, page 19.

This configuration includes the following elements:

Element	Explanation
the parameters in the preceding command (**Set-MsolDomainAuthentication**)	see Example of PowerShell command flow:, page 17
a reference to a Sign&go authentication scheme of type **Kerberos**, for the authentication by Web service	▶ You must configure an authentication scheme of this type in order to process the authentication requests for the **/WSFed/ windowstransport** Web service (see Example of configuration of a Sign&go authentication scheme of type Kerberos for the WS-Federation IdP, page 21).
a reference to a Sign&go authentication scheme of type **Identifier/password** for the authentication by Web service	▶ You must configure an authentication scheme of this type in order to process the authentication requests issued from Microsoft towards the Web service **/WSFed/ usernamemixed**. ***Note :*** *Microsoft transmits the* `UserPrincipalName` *attribute as login, instead of the Windows user name. You must then configure this authentication scheme accordingly. To do so, use the description of a directory which uses the LDAP* `UserPrincipalName` *attribute as user identifier.*
parameters set by the Microsoft Office 365 platform	These parameters are indicated in **orange** in the Example of configuration of the WS-Federation IdP for Microsoft Office 365 (page 19).
the attribute consolidation script	This script positions the attributes below in the attributes of the assertion which is sent to Microsoft: ▫ `ImmutableID`: 128-bit integer, coded in Base64. ▫ `UPN`: user's e-mail address. ***Note :*** *These attributes define the user and must exist in the assertion which is sent to Microsoft.* *In the Example of configuration of the WS-Federation IdP for Microsoft Office 365 (page 19), we retrieve these attributes from the user's Active Directory attributes,* `userPrincipalName` *and* `objectGUID`.

3.2.2.1 Configure the WS-Federation IdP for Microsoft Office 365

The configuration of the partners in the federation is performed in the Sign&go administration console, in: ***Partners*** *menu >* ***WS-Federation IdPs****.*

Tableau 1 • 1 : Example of configuration of the WS-Federation IdP for Microsoft Office 365

Section	Parameter	Value
-	**Partner name**	default ***Note :*** *You must reserve this name for Microsoft Office 365.*
	Description	WS-Federation test
Partner Issuer	**Our issuer**	http://air.airline.com/signandgo
	Partner issuer	**urn:federation:MicrosoftOnline**
Signature certificate	**Certificates keystore path**	**C:\my_certtests\mycert.jks**
	Keystore password	**********
	Alias of the signature certificate	myCertAlias
URLs	**Assertion consumer URL**	**https://login.microsoftonline.com/login.srf**
	Authentication URL	https://air.airline.com/auth/chooseschema.jsp
	Error page URL	https://air.airline.com/auth/error.jsp
NameId	**SAML format of user's identifier**	**urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified**
	Mapped LDAP attribute	`objectGUID`
	Binary formatted attribute	Check box selected
SAML conditions	**Audience restriction**	**urn:federation:MicrosoftOnline**
Web services authentication	**Name of the associated login/password scheme**	lp2
	Name of the associated Kerberos scheme	KRB ***Note :*** *see* *Example of configuration of a Sign&go authentication scheme of type Kerberos for the WS-Federation IdP, page 21*

Section	Parameter	Value

Script	**Script**	`// this script adds the UPN and ImmutableID attributes required for`
		`// an Office 365 token that is adapted for an AD directory`
		
		`var request = getRequest();`
		`var guid = getToken().getGUID();`
		`var user = guid.getUserID();`
		`var attributes = token.getUserAttributes();`
		`// getting UPN`
		`var userPrincipalName = attributes.getValue("userPrincipalName");`
		`// getting immutableID`
		`// script portion applicable to an Active Directory directory`
		`// where the immutableID is generated in binary format`
		`var objectGUID = attributes.getValue("objectGUID;binary");`
		`// Conversion of the attribute into a String to get the length`
		`// of the table`
		`var st = new java.lang.String(objectGUID);`
		`var chaineOriginale= "" + new java.lang.String(objectGUID,0,st.length());`
		`var crp = new Crypto();`
		`var encode = crp.base64Encode(chaineOriginale);`
		`// add immutableID to assertion`
		`wstrust.addAttribute("http://schemas.microsoft.com/LiveID/Federation/2008/05/ImmutableID", encode);`
		`// add userPrincipalName to assertion`
		`wstrust.addAttribute("http://schemas.xmlsoap.org/claims/UPN", userPrincipalName);`

	Script Debugging	-

*The parameters highlighted in **orange** correspond to the parameters set by the Microsoft Office 365 platform.*

*For more information on the configuration of the WS-Federation IdP in Sign&go, refer to the **Help** of the Sign&go administration console.*

3.2.2.2 Configure a Kerberos authentication scheme for the WS-Federation IdP

In this section:

3.2.2.2.1 Example of a Kerberos authentication scheme for the WS-Federation IdP

*The following example presents an example configuration of a Sign&go authentication scheme of type **Kerberos** for the WS-Federation IdP which has been previously configured (see Example of configuration of the WS-Federation IdP for Microsoft Office 365, page 19).*

In this example:

- *We use two different domain names (an external and an internal names) to identify:*
  - *On the one hand, the KDC in the intranet*
  - *And on the other hand, the "service principal name" which represents the Kerberos resource which rich clients access.*
- *The KDC is identified by its DNS name in the internal domain (**.lan**).*
- *In terms of Kerberos, the "service principal name" is the protected resource to which the rich client accesses in HTTP (therefore with an external resource name ending with **.com**) that is defined in the internal domain with **.lan**:*

  HTTP/air.airline.com@AIRLINE.LAN

Tableau 1 • 2 : Example of configuration of a Sign&go authentication scheme of type Kerberos for the WS-Federation IdP

Section	Parameter	Value
-	**Scheme name**	KRB ***Note :*** *Name pointed by the parameter **Name of the associated Kerberos scheme** that is configured in the IdP (see Table 1 • 1 - Example of configuration of the WS-Federation IdP for Microsoft Office 365, page 19).*
	Scheme description	Kerberos
	Authentication type	Kerberos (API Authentication agent)
	Level of confidence	1
	Directories to use	AD-AD

Section	Parameter	Value
Kerberos configuration	**KDC server**	air.airline.lan
	Kerberos realm	AIRLINE.LAN
	Activate Kerberos debugging	Check box selected
	Service Principal Name	HTTP/air.airline.com@AIRLINE.LAN ***Note :*** *Note that the resource name ends with* ***.com*** *and that the domain name ends with* ***.lan****.*
	Service password	********** ***Note :*** *It is the user's password* ***snguser_srv*** *(see Link the Kerberos authentication scheme with the domain's KDC, page 22).*
Script for consolidating token's application parameters	**Script for consolidating token's application parameters**	`var request = authenticationschema.getRequest();` `var token = authenticationschema.getToken();`
	Activate script debugging	Check box selected

3.2.2.2.2 Link the Kerberos authentication scheme with the domain's KDC

To link the Sign&go authentication scheme of type **Kerberos** with the domain's KDC, proceed as follows:

1. Create a user in Active Directory corresponding to the Kerberos service associated with the Sign&go IdP.

   Example: **snguser_srv**, with a "mypass" password which never expires.

2. On the domain controller, to associate a Kerberos service with this user, execute the following standard Windows command lines:

```
setspn -u -s HTTP/air.airline.com@AIRLINE.LAN snguser_srv

ktpass /princ HTTP/air.airline.com@AIRLINE.LAN /pass mypass /mapuser
snguser_srv@airline.lan
```

3.2.3 Create federated users on Microsoft Office 365

A user becomes a federated user once you have registered him (or her) with the Microsoft Office 365 site, with the domain name of the Office client (for example: jdoe@airline.com is a federated user on the airline.com domain).

When you register a user with the Microsoft Office 365 site, using the Microsoft Office 365 administration interface, this person is a “managed” user by default, with no link with the Office client Web site (for example: the **admin** user registered with this method for the airline.com client is given the following qualified name: admin@airline.onmicrosoft.com).

This method is inappropriate for the creation of a “federated” user, whom the Sign&go IdP will be in charge of authenticating.

You can use several methods:

Method	Explanation
Use the **PowerShell administration interface** of the site with the appropriate commands	For didactic purposes, we describe the creation of users with the help of PowerShell commands, in this section. ***Note :*** *You can use this “basic” method for tests or for specific needs however, to create and maintain a large number of users, this method would have to be enhanced.* ▶ To create a federated user on Microsoft Office 365 using the PowerShell interface, you have to: **1.** Create the user in the local Active Directory (page 24) **2.** Create the user on the Microsoft Office 365 platform (page 25)
Use the **Ilex Meibo connector** for Microsoft Office 365	The **Ilex Meibo** connector for Microsoft Office 365 is documented in another guide. ***Note :*** *For more information on Meibo’s Sign&go connector, the **Meibo Advanced User Guide** (chapter: **Provisioning** > **Targets** > **Office 365 target**) is available on request. To obtain it, please contact Ilex Customer Support.*
Use the **Microsoft DirSync tool**	You can download this tool from the Microsoft Office 365 site, using the site’s administrator account. This tool lets you create and synchronise users from the client directory, with the Microsoft Office 365 directory.

3.2.3.1 Create the user in the local Active Directory

To create the user in the local Active Directory, proceed as follows:

1. Using the standard Microsoft tool for managing Active Directory accounts, create user “John Doe” in the local Active Directory of the **airline.lan** domain.
2. Configure the e-mail address of the Microsoft Office 365 account of user “John Doe”: jdoe@airline.com, then complete the `userPrincipalName` attribute of this user with the same e-mail address.
3. Display the main attributes of user “John Doe” with the help of the PowerShell script below.

   This script summarises the creation operation and provides the `ImmutableID` value which Active Directory has assigned the user:

```
$samID = "jdoe"  // Windows identifier created for the user
$searcher = [DirectoryServices.DirectorySearcher]
"(samaccountname=$samID)"
$user = $searcher.FindAll()|%{$_.GetDirectoryEntry()}
$guid = [Guid]($user.Properties["objectGUID"][0])
$base64 = [System.Convert]::ToBase64String($guid.ToByteArray())
$lastname = $user.Properties["sn"]
$firstname = $user.Properties["givenname"]
$displayName = $user.Properties["displayname"]
$upn = $user.Properties["userPrincipalName"]
$lastname = $user.Properties["sn"]
$firstname = $user.Properties["givenname"]
$displayName = $user.Properties["displayname"]

Write-Host "Firstname: " $firstname
Write-Host "Lastname: " $lastname
Write-Host "UPN: " $upn
Write-Host "ImmutableId: " $base64
```

   The output attribute values are as follows:

Attribute	Value
Firstname	John
Lastname	DOE
UPN	jdoe@airline.com
ImmutableID	a1zi9FB84kizjq9dOS0xiQ==

To complete the creation of the federated user on Microsoft Office 365, with the PowerShell interface, you must now *Create the user on the Microsoft Office 365 platform (page 25).*

3.2.3.2 Create the user on the Microsoft Office 365 platform

Before creating the user on the Microsoft Office 365 platform, you have to Create the user in the local Active Directory (page 24).

To create the user on the Microsoft Office 365 platform, proceed as follows:

1. First, retrieve the Microsoft Office 365 license identifier using the following command line:

```
Get-MsolAccountSku
```

   This command returns a Microsoft Office 365 license identifier such as: **airline:LITE_PACK_P2**.

2. Use the license identifier as well as the values previously collected about the user (see Create the user in the local Active Directory, page 24 - step 3) to define the user creation command, then run the following command:

```
New-MsolUser -UserPrincipalName jdoe@airline.com
   -DisplayName 'John DOE'
   -FirstName John
   -LastName DOE
   -LicenseAssignment airline:LITE_PACK_P2
   -UsageLocation FR
   -ImmutableId a1zi9FB84kizjq9dOS0xiQ==
```

3.2.4 Delete a Microsoft Office 365 user

- In a test phase, if you want to unsubscribe user "John Doe" and remove him from the bin in order to be able to re-create this user from scratch, run the following command lines:

```
Remove-Msoluser -UserPrincipalName jdoe@airline.com
Remove-Msoluser -UserPrincipalName jdoe@airline.com -RemoveFromRecycleBin
```

3.3 Implement the connection

In this section:

3.3.1 Implement the connection for rich clients

To implement the connection for rich clients, you do not have any specific operation to carry out besides those above listed.

- *To connect, user "John Doe" launches his Outlook client, creates his Outlook account, jdoe@airline.com, and is then connected to the Office 365 Exchange server.*

3.3.2 Implement the connection for light clients

The user can use the **SP-initiated** or **IdP-initiated** modes:

If the user uses the...	... then the implementation of the connection is as follows:
SP-initiated mode	1. The user launches his Web browser and connects to the Microsoft Online site using his federated login, jdoe@airline.com, without entering his password. 2. The Microsoft Online site redirects the user to the Sign&go IdP site. 3. If the user has not already authenticated to Sign&go, according to the authorisation rules that you have configured in the Sign&go administration console, the Sign&go IdP may request the user to authenticate. 4. Then, Sign&go redirects the user to Microsoft Office 365 which automatically authenticates him.
IdP-initiated mode	1. In order to access the Sign&go IdP site, the user enters the specific URL below in the Web browser: https://air.airline.com/WSFed/ls/default?wa=wsignin1.0&wtrealm=urn:federation:MicrosoftOnline 2. The Sign&go IdP invites the user to authenticate locally if he is not authenticated already, and the user is then authenticated on Microsoft Office 365.

3.4 Debug connection problems

In this section:

3.4.1 Internal debugging tools

The following internal debugging tools are available:

Internal Debugging tools	Explanation
Loggers of the Sign&go Web application	These loggers can be found under logger **ilex.signandgo.WSFed**. ▶ You can activate them in the **/WSFed/WEB-INF/client.xml** file.
Logger of the Sign&go security server	This logger is the **logsngsrv.wstrust** logger.
Debug mode of the Kerberos authentication scheme	Concerning the Kerberos authentication, you can configure the Sign&go authentication scheme of type **Kerberos** in debug mode (see Example of configuration of a Sign&go authentication scheme of type Kerberos for the WS-Federation IdP, page 21, parameter **Activate Kerberos debugging**).

3.4.2 External debugging tools

The following external debugging tools are available:

External Debugging tools	Explanation
Microsoft tool https://testconnectivity.microsoft.com/	This tool helps debug the connection problems with Microsoft Office 365. It simulates and tests operations in login/password mode (in other words, in the context of user connections from rich clients outside of the domain) or in passive federation mode. It then delivers a diagnostic on connection problems such as considered from the Microsoft Office 365 platform. ▶ To run the test, select: **Office 365** tab > section **Office 365 General Tests**> **Office 365 Single Sign-On Test** radio button, then complete the authentication form. ***Note :*** *This unique session opening test validates your ability to open a Microsoft Office 365 session with your local identification information.*
"Wireshark" analysis tool	This tool helps debug connections of users operating with rich clients on the domain, thus via Kerberos authentications. ▶ Place this tool on the server hosting the Sign&go IdP in order to view: ◘ IdP flows ◘ Kerberos flows on the KDC ***Note :*** *The flows towards the Sign&go IdP are in SSL mode therefore, they are encrypted. You can introduce the SSL key of the server in the* ***Wireshark*** *tool to decrypt these flows.* *The flows that you can observe on the Sign&go IdP are the requests and responses to the queries on metadata, passive federation and the Web services* ***WSFed/usernamemixed*** *and* ***/WSFed/windowstransport****, that correspond to the authentications in login/password mode and Kerberos mode, respectively.* ***Note :*** *The absence of requests on entry points means that you have probably made an error in the configuration of the URLs transmitted to Microsoft Office 365 via the* ***Set-MsolDomainAuthentication*** *command. The "KO" SOAP responses that the Sign&go IdP issues usually include a diagnostic of the error such as considered from Sign&go.*