# Description

### This module enables Kerberos based Single Sign On in your application. Very useful in Windows Active Directory environments, since this protocol supports Integrated Windows Authentication. As a result, users logged on to the windows domain do not have to enter their credentials to log in to your application.

# Features and limitations

* Kerberos must be enabled, in an Active Directory environments older protocols, such as NTLM, are not supported.
* Computers must be connected and logged on to the domain in order to automatically log in.
* The Mendix server must be a member of the same AD domain as the users in order for SSO to work. This also means this module won’t be usable in Mendix cloud environments.

This module combines well with the LDAP synchronization module, which can be used to synchronize the userbase with the Active Directory Domain server.

The working of this module is very dependent on the network and domain environment. The working of this module depends on both the local machine security policies and the domain security policies. In order to setup this module in the company domain, the knowledge of an experienced domain administrator is required!

# Installation

We made installation as easy as possible, but creating Kerberos keys is not trivial and requires domain administrator rights. Please follow this manual carefully. This manual applies for Active Directory domains only, but may be used as guide for other environments.

### For debugging purposes, access to the security event log of the domain controller is required. Most relevant information is logged at the domain controller and not at the client (Mendix) application.  More details for the most important steps will be given below.

1. Import the module into your project
2. Configure the model to enable Single Sign On support
3. Create a service user and keytab file on the domain server
4. Set the Kerberos details in the properties file
5. Synchronize the userbase (e.g. using the LDAP module)
6. Enable Integrated Security support in all user browsers

### Enable Single Sign On support in the model

After importing the module, the following steps should be performed in the model.

First, the WinSSO.startmicroflow should be set as after startup microflow.

Second, the module supplies a replacement for the default *login.html*. Rename the existing *login.html* file in your theme to *login-default.html*, and copy the new *login.html* file found in <project dir>/resources/winsso/ to your theme. This login.html will redirect users to the */sso/* page, which will trigger the single sign on procedure.

Optionally, you can configure your webserver to use the *login.html* page as the default page instead of *index.html*. In this way, browsing to *<yourdomain>/* should result serving *<yourdomain>/login.html*. Refer to the documentation of your webserver to accomplish this. This will make sure users are automatically logged in when navigating to your app, but a manual login will require navigating to *<yourdomain>/login-default.html* manually.

Next, decide on how you want to deal with unknown users, i.e. when a user is authenticated through SSO, but unknown in the application. By default, an HTML page with an error message is shown. To use this behavior, copy the *unkownuser.html* page in the resources folder to your theme, and set the *UnknownUserPage* constant (see below) to its default value.

The IndexPage constant should be set to the file that loads the Mendix client. Usually this will be 'index.html', but in specific cases it might be something else. The user will be redirected to this page on a successful login. If you don’t know you need this, don’t change it.

The *UnknownUserPage* constant decides where a user that is successfully authenticated will be redirected. By default, it will redirect to *unkownuser.html*, which shows an error message. Another option is to redirect to a deep link the opens an account creation form. Please see the deep link module for more information.

The final step is to configure the files <project dir>/resources/winsso/sso.propertiesand .keytabfile in the same directory. This will be explained in the next paragraph.

### Configure Kerberos

Logon to the domain server with an administrator account and add an additional user account to the active directory server (figure 1). Its username (for example MendixKerberos) is required in the next step. The password is not needed furthermore and can be randomized.

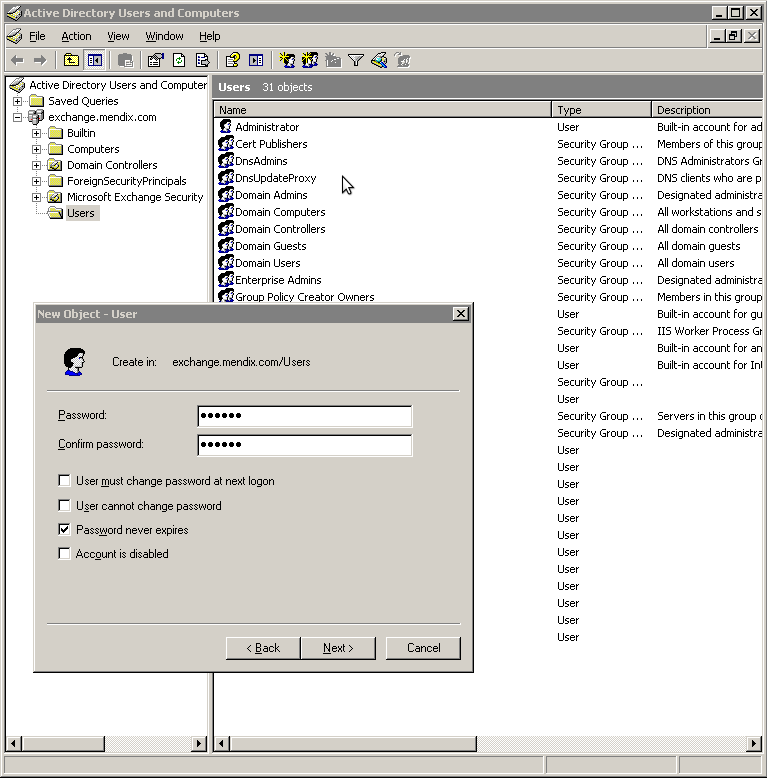
**

figure 1. Creating a user account which will be used as service account for authenticating

Kerberos requires to define a service principal (SPN) for the web server that hosts the Mendix application. For this service we create a .keytab file which can be used to authenticate the Mendix application. On the domain controller open a console and enter the following command (if necessary, make sure Windows Support Tools are installed):

ktpass.exe -princ <protocol>/<hostname of server>.<domain in lowercase>@<domain in upercase> -mapuser <username from previous step> +rndPass -out <.keytab filename> -ptype KRB5\_NT\_PRINCIPAL

 for example:

ktpass.exe -princ HTTP/mxapp.example.local@EXAMPLE.LOCAL -mapuser MendixKerberos +rndPass -out mxapp.keytab -ptype KRB5\_NT\_PRINCIPAL

The output should look a bit like:

C:\Documents and Settings\adm> ktpass.exe -princ HTTP/mxapp.example.local@EXAMPLE.LOCAL -mapuser MendixKerberos +rndPass -out mxapp.keytab -ptype KRB5\_NT\_PRINCIPAL  
Targeting domain controller: dc1.EXAMPLE.local   
Successfully mapped HTTP/mxapp.example.local to MendixKerberos.   
Key created.   
Output keytab to mxapp.keytab:   
Keytab version: 0x502   
keysize 83 HTTP/mxapp.example.local@EXAMPLE.LOCAL ptype 1 (KRB5\_NT\_PRINCIPAL) vno 2 etype 0x17 (RC4-HMAC) keylength 16 (0xb5430ce6896bdecb8f02a8e28baf6ccd)

Put the generated keytab file in the resources folder inside the project directory (<project directory>/resources/winsso/). This way, it will be automatically included in deployments to the Mendix server.

Note that you have to use the same address in the browser as specified in this command to navigate to the application, in order make Single Sign On work. In the example above, this would be http://mxapp.example.local/. In most cases, omitting the domain (http://mxapp/) or using https will work as well. Accessing the server through a different DNS name will cause the SSO to fail.

Next, the <project directory>/resources/winsso/sso.propertiesfile should be adjusted according to this keytab. An example file is shown below, replace the values with those you used in the ktpass command.

#the domain used to sign on to   
domain = example.local   
#ip adress of the domain controller   
active\_directory\_server = 10.140.10.1   
#service name of this server, as it is known by the kerberos ticket service   
kerberos\_servername = mxapp   
#the keytab file (should be in the resources/winsso/ folder)   
kerberos\_keytab\_file = mxapp.keytab  
#protocol for which the keytab was generated, either http or https   
kerberos\_protocol = http  
#debug information printed? (only to the shell)   
debug = false

### Next steps

Your application is now ready to be deployed. However, you have to decide on how user accounts will be created in your application. For example, when a user with Windows Active Directory Account John@yourdomain.comtries to log on to your application, the user John needs to exist in the user database of your application. The password for this user can be random, it will not be used (unless using the work around described below). The LDAP synchronization is a very useful module to synchronize users with the Active Directory. Furthermore Active Directory based authentication must be supported by the browser, see the next step.

If you want to log in without using integrated windows authentication, go to <application\_URL>/login-default.html and provide the credentials manually. This should also be done in case of local deployment.

### Preparing the browser for Single Sign On

Most browsers have integrated windows authentication turned off by default. This document describes how to enable integrated windows authentication. Note that these settings can be rolled out centrally in a well configured domain.

#### Internet Explorer

Go to Tools > Internet Options > Security tab, Select local intranet > Sites > Advanced and add the URL of the application.

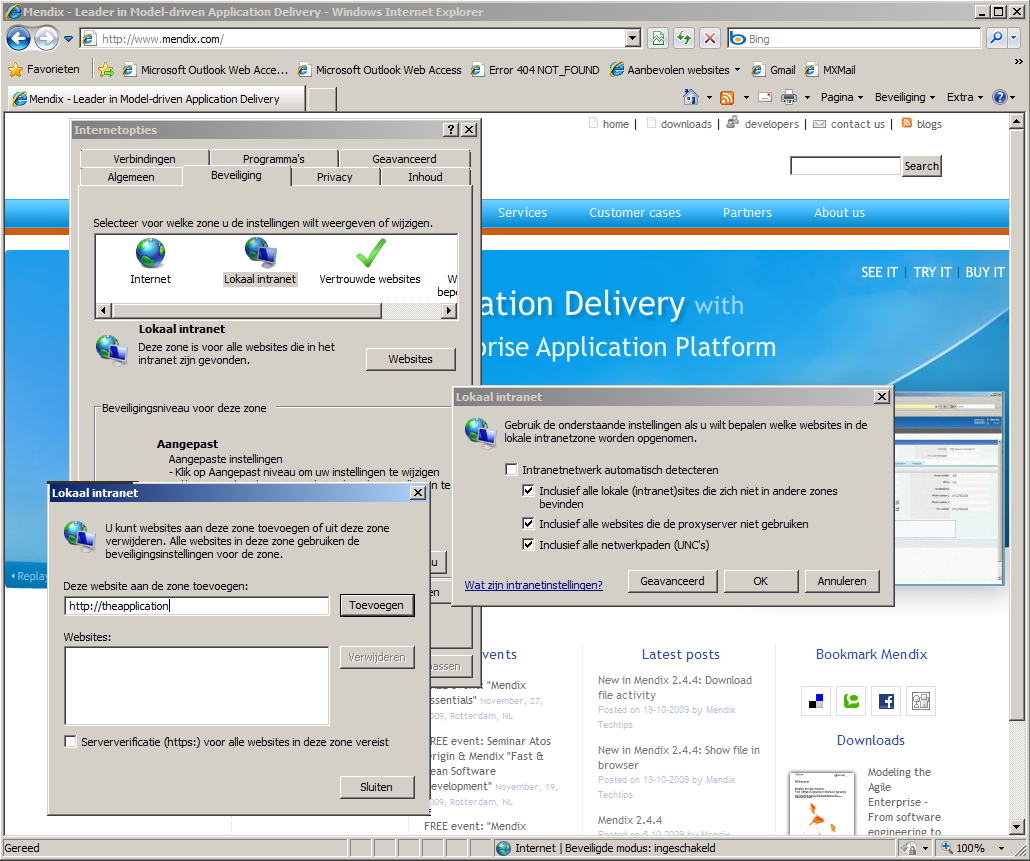
**

figure 2. Enabling Single Sign On in Internet Explorer

Go back to the security tab and press custom level. Scroll down to the user authentication section, and select for Logon > Automatic login only in Intranet zone.   
Close the custom level window and select the advanced tab. Make sureEnable Integrated Windows Authentication is checked.

#### Firefox

In Firefox go the address: about:config (accept the security warning if any).   
Set the network.negotiate-auth.delegation-uris and network.negotiate-auth.trusted-uris properties to the value 'http://,https://'. (more specicific URLs might be used, for example 'https://mendix.com') .

#### Chrome

Chrome uses the same settings as Internet Explorer, which are the Windows system settings. Follow the instructions for IE to configure Chrome for SSO.

# Frequently Asked Questions

#### *How can I access the server through another (additional) URL than I specified in the ktpass command?*

#### To make this work, you’ll have to link the SPN of the new URL to the Mendix service user, and also include the new URL in the local intranet zone of your clients. To link the SPN, execute this command in a command-line on the domain controller:

setspn –S <protocol>/<new url> <service username>

 for example:

setspn –S HTTP/otherurl.example.local -mapuser MendixKerberos

You can use the other options of the *setspn* command (-Q, -L and -X) to check the results. Note that this might not work when using a URL in a different domain from the original URL, and will not allow users from that other domains to log in.

#### *Error: Clock skew too great (37)*

The time or time zone of the application server does not match the time or time zone of the active directory domain controller

#### *Kerberos error: invalid token Negotiate TlRMTVNTUAABAAAAB4IIog[…]== (GSSHeader did not find the right tag), falling back to alternative authentication mechanism*

Kerberos is not enabled, and the browser used an old fashioned NTLM token. Please follow the browser installation instructions at this page. This error can also occur when the SPN used in the ktpass command does not precisely match the URL used to access the application. Either run the ktpass command using the right URL, or access the application via the URL used in ktpass.

#### *Error: Unable to create session: Single Sign On unable to create new session: Login succeeded, but user not found: gebruiker1*

Single sign on succeeded, however the user logged on the domain does not exist in the application