

Novell Access Manager authentication class for OpenID authentication

(Requires NovellAccessManager 3.1 SP1 IR1 or later)

Introduction:

This article describes the steps to deploy and configure a new authentication class that performs OpenID authentication from Novell AccessManager. This authentication class can be deployed on 3.1 SP1 IR1 or later. The OpenID authentication is achieved by using the opensource java OpenID library available at <http://code.google.com/p/joid/> . Once deployed, you can configure the authentication class properties to identify the user at local store.

Steps:

1. Download OpenID libraries
 - a) Download joid-1.1.war from <http://joid.googlecode.com/files/joid-1.1.war>
 - b) Extract the war file into a directory
 - c) /usr/java/jdk1.5.0_13/bin/jar xvf joid-1.1.war (check the jdk version you have on your system)
 - d) The jar files will be located in the <extracted dir>/WEB-INF/lib/ . Copy joid.jar, tsik.jar to /var/opt/novell/tomcat5/webapp/nidp/WEB-INF/lib directory on your Identity server
2. Download Authentication Class and other binaries
 - a) Download the Novell AccessManager OpenID Authentication class binaries file from <http://www.novell.com/communities/files/OpenIDAuthClass.zip>
 - b) Unzip the downloaded file
 - c) Copy the openidclass.jar to /var/opt/novell/tomcat5/webapp/nidp/WEB-INF/lib directory on your Identity server
 - d) Copy the OpenIDResources_en.properties file to /var/opt/novell/tomcat5/webapps/nidp/WEB-INF/classes directory
 - e) Copy the openIdlogin.jsp to /var/opt/novell/tomcat5/webapp/nidp/jsp directory on your Identity server
3. Deploy and configure /var/opt/novell/tomcat5/webapps/nidp/WEB-INF/web.xml
 - a) Edit the web.xml and add openIdlogin.jsp to the param-value of publicAccess init-param of nidpJspFilter

```
<filter>
  <filter-name>nidpJspFilter</filter-name>
  <display-name>NIDP Jsp Filter</display-name>
  <description>The NIDP server JSP filter. Enforces authentication and handles
clustering.</description>
  <filter-class>com.novell.nidp.servlets.filters.jsp.NIDPJspFilter</filter-class>
  <init-param>
    <param-name>publicAccess</param-name>
    <param-
value>main.jsp;err.jsp;err2.jsp;login.jsp;nmaslogin.jsp;logoutSuccess.jsp;banner.jsp;nav.jsp;menus.jsp;foo
ter.jsp;content.jsp;cards.jsp;title.jsp;error.jsp;curcard.jsp;createacct.jsp;x509err.jsp;openIdlogin.jsp</param
-value>
  </init-param>
</filter>
```
 - b) Add the openid filter to the web.xml

```

<filter>
<filter-name>OpenIdFilter</filter-name>
<description>This filter (for Consumer side) automatically parses OpenID responses and sets the
user's identity in the session.</description>
<filter-class>org.verisign.joid.consumer.OpenIdFilter</filter-class>
</filter>

```

c) Add the openid filter-mapping to the web.xml

```

<filter-mapping>
<filter-name>OpenIdFilter</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>

```

4. Create and configure a new authentication class

a) Create a new Authentication class idp-server → edit → local → classes → new

Create Authentication Class

Step 1 of 2: Specify name and java class.

Display name:

Java class:

Java class path:

Give a descriptive name of your choice, Select Other for Java Class and for the Java class path, type com.novell.nidp.authentication.local.openid.OpenIdClass
Click Next

Create the following properties

Create Authentication Class

Step 2 of 2: Specify properties.

[New](#) | [Delete](#)

<input type="checkbox"/> Name	Value
<input type="checkbox"/> OpenIdProviderUrls	myopenid.com;someotheropenid.com
<input type="checkbox"/> UserMappingType	LDAPATTRMAP
<input type="checkbox"/> LdapMappingAttribute	carLicense
<input type="checkbox"/> AutoProvision	true

Click Finish.

The following table gives more details about the different properties.

Properties Name	Optional/Mandatory	Possible values
OpenIdProviderUrls	At least one string should be present	This is a semi-colon (;) separated list of strings. The OpenID url that user enters during login process must contain one of the strings as a subset of the OpenID url. For example, user enters https://user123.myopenid.com . Administrator configures, “myopenid.com” or “.myopenid.com” as the provider url.
UserMappingType	Mandatory	NONE or LDAPATTRMAP Notes: NONE - The authentication class does not try to map the OpenID user to a user in local userstore. The Identity of the user will remain as the OpenID url at the Identity Server. This method may not be really useful if you are using AccessManager roles for Authorization as the NAM policies require an LDAP user to be associated with the authentication. LDAPATTRMAP – This authentication class tries to map the OpenID user to a user in local userstore whose ldap attribute' value is equal to the OpenID url that is just authenticated. The name of the attribute is specified by the “LdapMappingAttribute” property
LDAPATTRMAP	Mandatory if UserMappingType is LDAPATTRMAP	This Attribute should already be in the LDAP schema of the user. And this attribute should have read permission.
AutoProvision	Mandatory if UserMappingType is LDAPATTRMAP	If the administrator does not want to provision the attribute values with the OpenID URLs, then AutoProvisioning can be done. This is similar to the AutoProvisioning feature of X509 Authentication class. When AutoProvision=true, and if the LDAP attribute mapping is failed, then user is prompted to enter his local userstore credentials for authentication. When the local authentication is successful, the LdapAttribute is provisioned on that user with a value of his/her OpenID URL. Next time, when the OpenID authentication happens from the same URL, LDAP attribute mapping will succeed.

5. Create a contract and authentication method to use this authentication class
 - a) Create a new authentication method (idp-server → edit → local → Methods → new)

Create Authentication Method

Configuration

Display name:

Class: ▼

Identifies User

Give a descriptive name and choose the OpenID-Auth-Class as the class. Configure the remaining fields according your environment. Click Finish

- b) Create a new authentication contract ((idp-server → edit → local → Contracts → new))

Create Authentication Contract

Step 1 of 2: Configuration

Display name:

URI:

Password expiration servlet:

Allow user interaction

Authentication Level:

Satisfiable by a contract of equal or higher level

Satisfiable by External Provider

If you add more than one X509 method, only the first one will be used and it will automatically be moved to the top of the list.

Methods:

Available methods:

-
-
-
-

⬆ ⬇

Choose a descriptive name, URI and choose the OpenID-Auth-Method in the authentication methods list. Click Next, configure the next section, click Finish and apply the configuration changes to the Identity Server.

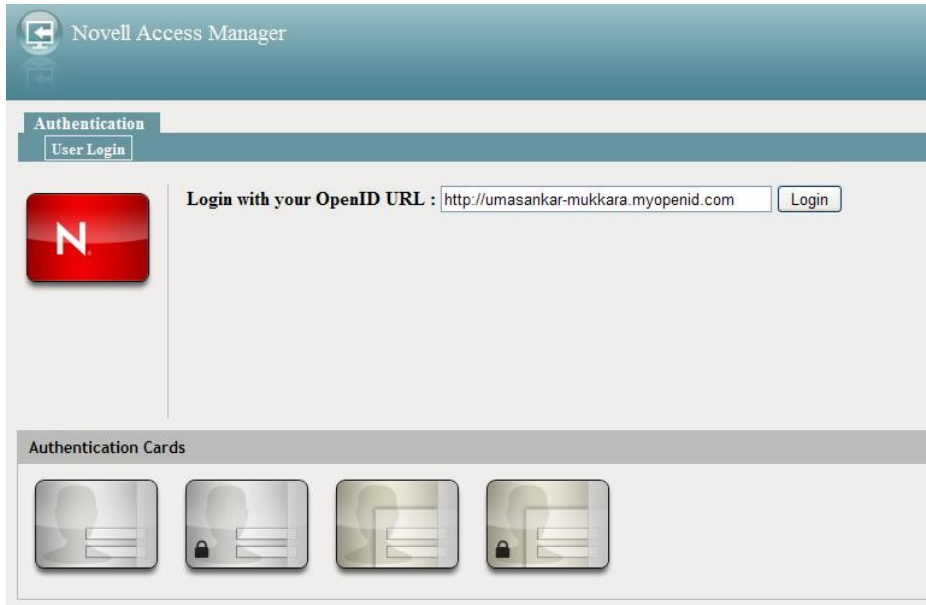
6. Localization

You can localize the messages by creating localized files of `OpenIDResources_en.properties` located in `/var/opt/novell/tomcat5/webapps/nidp/WEB-INF/classes` directory.

Appendix

User Login Screen shots

Scenario 1: LDAP Attribute Mapping is NONE



The screenshot shows the Novell Access Manager user login interface. At the top, there is a header with the Novell logo and the text "Novell Access Manager". Below this is a section titled "Authentication" with a sub-tab "User Login". On the left, there is a red square icon with a white letter "N". To the right of the icon, the text reads "Login with your OpenID URL : http://umasankar-mukkara.myopenid.com" followed by a "Login" button. Below the "Authentication" section is a section titled "Authentication Cards" which contains four icons representing different authentication methods: a standard user card, a card with a lock icon, a card with a key icon, and a card with a lock and key icon.

Initial user authentication screen



The screenshot shows the myOpenID sign-in interface. At the top, there is the myOpenID logo. Below the logo is a green header with the text "SIGN IN". Underneath the header is a "Notice" section with a blue exclamation mark icon and a "Dismiss" link. The notice text reads: "You must sign in to authenticate to https://jjaimon-csb.dnsdhcp.provo.novell.com:8443/nidp as http://umasankar-mukkara.myopenid.com/". Below the notice, there is a "Username" field with the value "http://umasankar-mukkara.myopenid.com/" and a "Password" field. There is also a checkbox labeled "Stay signed in". At the bottom right, there are "Sign In" and "Cancel" buttons. At the bottom left, there are two links: "Sign in with an SSL certificate" and "I cannot access my account".

User is redirected to OpenID provider for authentication


Novell Access Manager N

Authentication Profile Welcome: <http://umasankar-mukkara.myopenid.com>

User Login

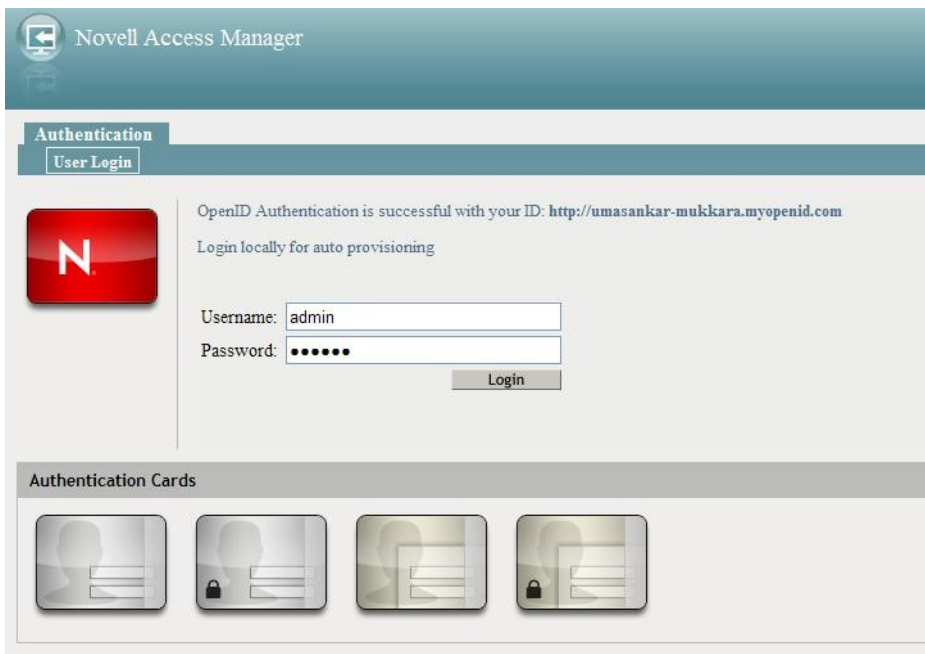
i Information: Your session has been authenticated and is valid for 60 minutes.

Authentication Cards

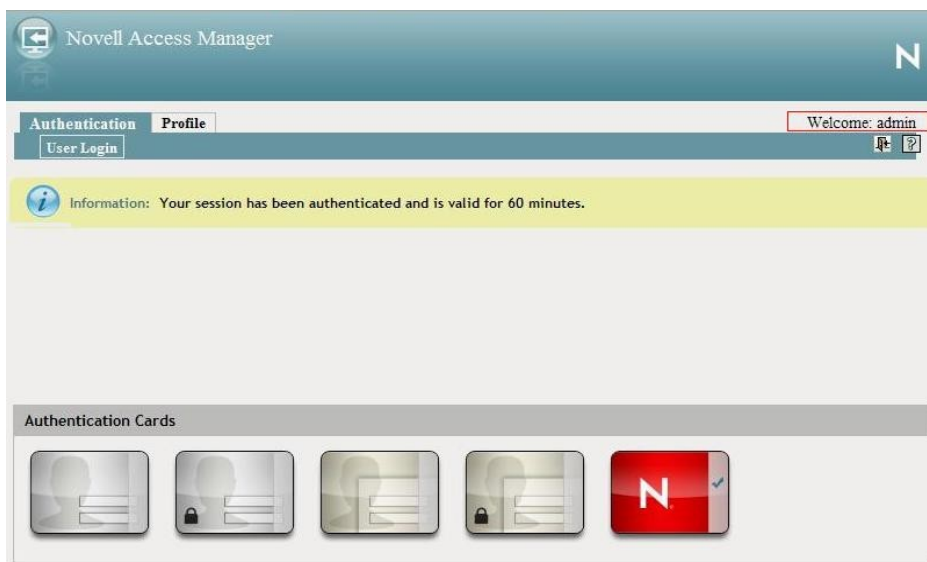


User is authenticated at Identity Server

Scenario 2: User Mapping Type is LDAPATTRMAP (AutoProvision = true)

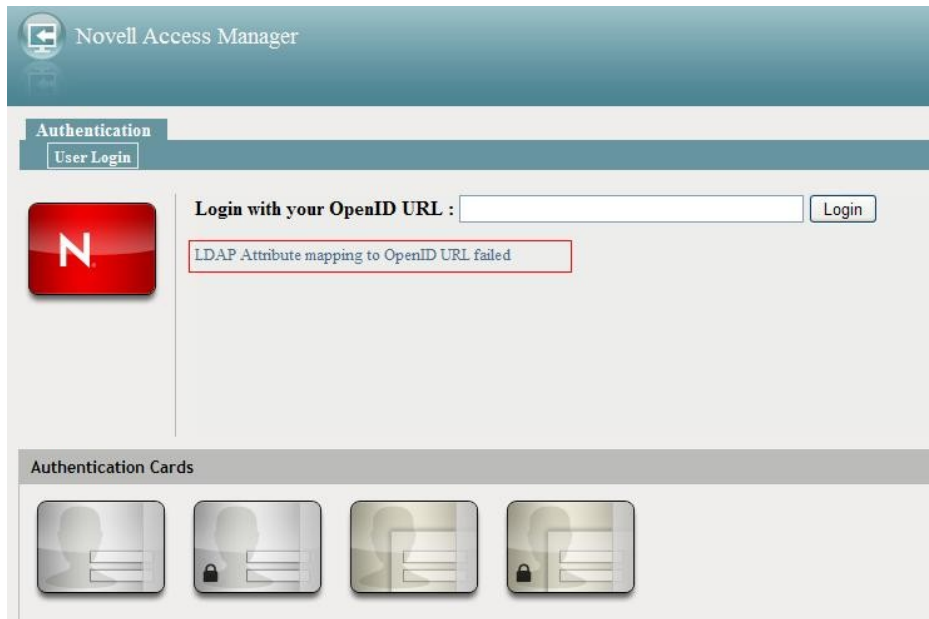


After OpenID authenticated, user is prompted for local credentials for auto provisioning of LDAP attribute.



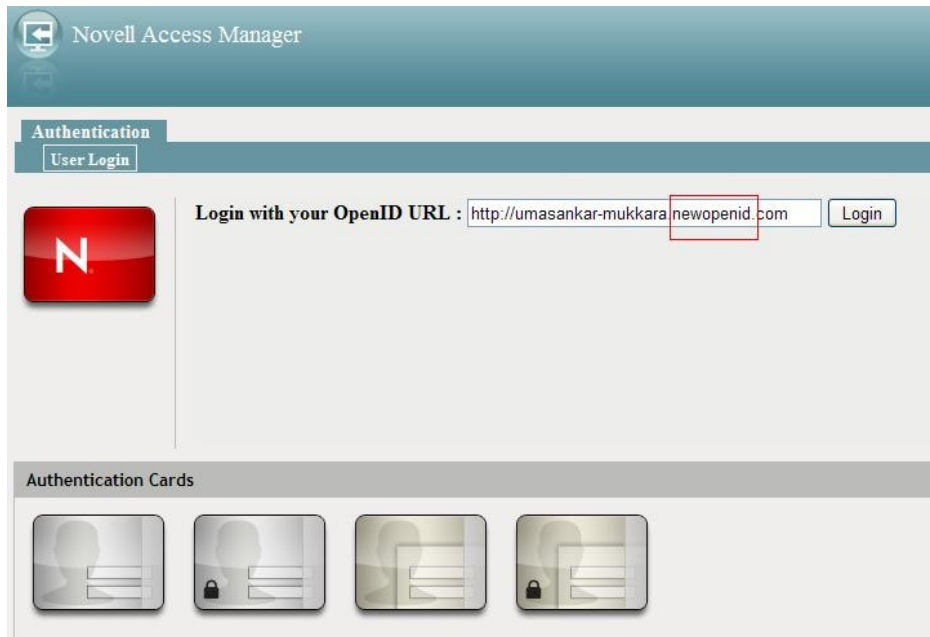
Auto provisioning is done. A mapping is done between admin user and the OpenID URL.

Scenario 3: User Mapping Type is LDAPATTRMAP (AutoProvision = False)

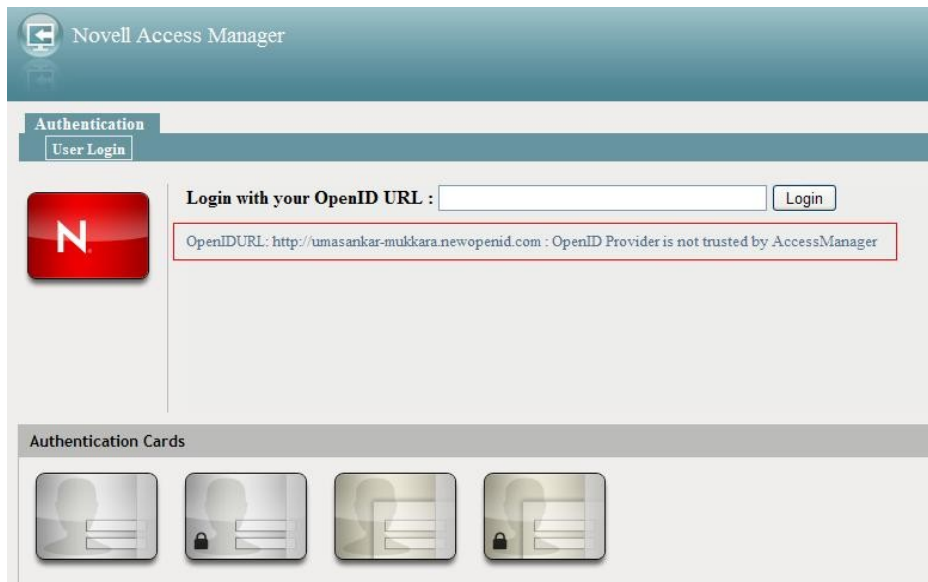


When auto provision is set to false and the attribute mapping could not be done, authentication into IDP fails.

Scenario 4: OpenID URL is not in the configured provider URLs



The OpenID url is not in the trusted URL provider list.



OpenID authentication is not even attempted if the URL is not in the provider list.