# **External Directory integration - ps-directory**

External directory is an external source for authenticating or authorization of users who are accessing EBX.

The ps-directory project supports EBX Directory, LDAP and SAML 2.0.

**EBX Concepts**

User – a person who want to access EBX

Role – an authorization to access or act on items in EBX

**Authentication:**

**EBX Directory**

The default way of logging into EBX. This option is supported with either LDAP or SAML 2.0

It can be combined with one of the two.

**LDAP**

A use access EBX login page provide his credentials and click on the login button. The user credentials are verified with the external directory, if they valid the user is granted access to the next step. If credentials are invalid user get the login page with a message stating that the user name or password are invalid.

**SAML 2.0**

Users login to EBX via an SSO URL. From that SSO URL the user is redirected to an IDP website for authentication. If the user is authenticated, he will be redirected to the EBX URL and be automatically logged in.

**Authorization:**

Once the user is authenticated the next step is authorization.

There are 3 options for authorization, EBX Directory, LDAP or SAML 2.0.

A user provides valid credentials and EBX computed what can he do as the next step after the login page. EBX query the directory for the user roles. if the user is assigned to roles then he gets access to the next step. if not, the user gets a message for access denied.

**LDAP**

EBX queries will be translated to LDAP quires to retrieve the user roles based on the user association with LDAP groups.

A role can be mapped to an LDAP group in 2 ways.

1. Groups are created in LDAP following a pattern, [prefix][EBX Role name][suffix] where prefix and suffix are optional
2. Groups already exist and the only way to map them is by creating fictive roles with the group names and make those new roles included in the related role.

**SAML 2.0**

EBX queries will be translated to a lookup into the USER roles that are brought back as part of the SAML response. The IDP need to be configured to return the groups for this feature to work.

### **Configuration requirements**

**LDAP**

**To Connect to the external directory**

1. To enable LDAP ebx.directory.factory=com.orchestranetworks.ps.customDirectory.LdapDirectoryFactory
2. An account to connect to the LDAP/ActiveDirectory server in one of the following options:
   1. The account that is used to run EBX/Tomcat have permissions and access to the

LDAP/ActiveDirectory

ebx.directory.ldap.bindDN=uid=admin,ou=system

* 1. A **Distinguished Name** and a password to connect to the LDAP/ActiveDirectory. **NOTE:** The password at this point will be stored in the configuration file in plain text.

ebx.directory.ldap.credential=secret

**Authentication:**

1. What is the baseDN. usually it is something like: dc=example,dc=com

ebx.directory.ldap.baseDN=dc=example,dc=com

1. How to search for a user: (&(objectClass=person)(uid={0}))
   1. {0} EBX replace this with the user login, on Active Directory we usually need to replace uid with "sAMAccountName"
   2. Person might be user or other type.

ebx.directory.ldap.search= (&(objectClass=person)(uid={0}))

**Authorization:**

1. How to search for a group: (&(objectClass=groupOfNames)(cn={2}))
   1. {2} EBX replace this with the computed role name (ebx.directory.ldap.membershipRole.environementPrefix + ebx role name + ebx.directory.ldap.membershipRole.environementSuffix)
   2. groupOfNames might be groupOfUniqueNames or group

ebx.directory.ldap.membershipRole=(&(objectClass=groupOfNames)(cn={2}))

1. How to search / check if a user belongs to a group : (&({2})(member=uid={0}))
   1. For the group: {2} - Ebx will replace this with the result from 4
   2. For the uid either the login name or the result from 3. It is possible we need something else instead of member.

ebx.directory.ldap.membershipFilter=(&({2})(member=uid={0}))

**Note:** For more information please refer to the ebx\_ldap\_configuration.properties.

**SAML 2.0**

**Authentication:**

1. To enable SAML 2.0 one of the following factories must be set:
   1. com.orchestranetworks.ps.customDirectory.LdapDirectoryFactory
   2. com.orchestranetworks.ps.customDirectory.CustomDirectoryFactory

ebx.directory.factory=

ebx.directory.enableSSO=true

ebx.directory.SSOClass=com.orchestranetworks.ps.customDirectory.sso.saml.SamlSSOCheck

1. To authenticate a user the following information is required:
   1. ebx.directory.sso.idp.url= IDP URL – usually a generated url by the IDP Server when an application is setup. This needs to support SAML 2.0 authentication requests
   2. ebx.directory.sso.idp.issuerid= provided by the IDP
   3. ebx.directory.sso.saml.user.email=User.email – assertion attribute name for email
   4. ebx.directory.sso.saml.user.firstname=User.FirstName – assertion attribute name for first name
   5. ebx.directory.sso.saml.user.lastname=User.LastName – assertion attribute name for last name.
   6. ebx.directory.sso.saml.user.roles=member-of – assertion attribute name for an attribute with multiple values for the groups names.
   7. ebx.directory.sso.sp.issuerid= a value that will identify EBX with the IDP server. EBX doesn’t provide this value but it can be any made up value as long as this value is the same in EBX configuration file and the IPD server configuration
   8. ebx.directory.sso.idp.isjksfile= indication is the file that indicated by “ebx.directory.sso.idp.certificate.file” is a jks file or not.
   9. ebx.directory.sso.idp.jkspassword= when a JKS file is used, this is the password to access the certificated in it
   10. ebx.directory.sso.idp.certificate.alias= when a JKS file is used,this is the alias to use to locate the certificate
   11. ebx.directory.sso.idp.certificate.file=a qualified name the can be either a certificate file or a JKS file (Java Key Store).
   12. ebx.directory.sso.sp.privatekey.file= a private key file to be used to decrypt encrypted assertion.

**Authorization:**

1. To use IDP authorization, groups needs to be defined on the IDP server and SAML response needs to be configured to return the user groups.
   1. ebx.directory.sso.saml.user.roles=[attribute name for returning the groups, e.g. member-of]
   2. ebx.directory.sso.useroles=true