

AfterthoughtSoft-Secure™

Enterprise Edition

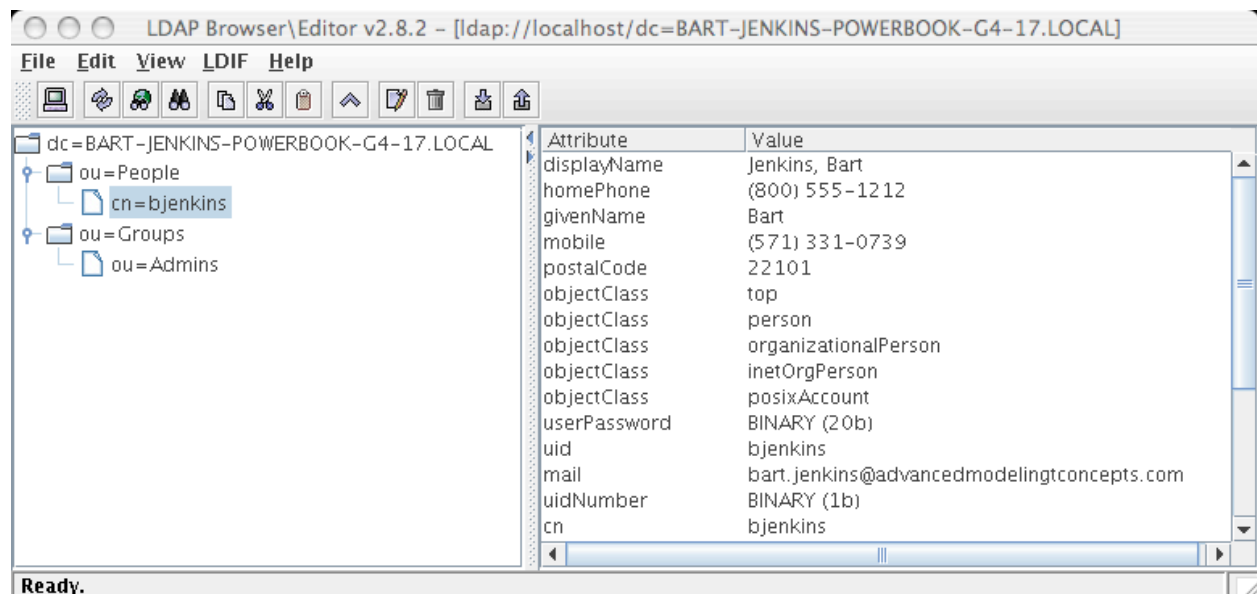
The Enterprise Edition adds to the power of the community and pro editions with the power to authenticate against industrial strength security realms-- specifically, any JNDI compliant directory service or Kerberos V.

JNDI compliant directory servers (NIS / LDAP)

Any JNDI compliant directory service can be used for authentication via the JAAS [JndiLogin-Module](#) from Sun. In this example, we add security to an application to authenticate against users in an [OpenLDAP](#) v3 directory.

LDAP is both a communication protocol and a data protocol. LDAP is based on the [RFC 1487](#), X.500 Lightweight Directory Access Protocol.

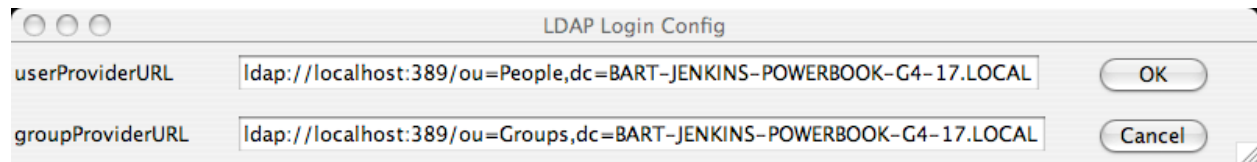
Imagine I have a user named 'bjenkins' whose login and contact information is stored in an OpenLDAP v3 repository. When browsing the directory with an LDAP browser, I might see a view like the following:



(This very nice JAVA based LDAP browser, written by Jarek Gawor, can be found [here](#))

To authenticate against an LDAP directory service, you need to provide:

- * The LDAP uri (ldap://)
- * The URL where the directory server resides (localhost)
- * The port (standard non secure port for LDAP is 389)
- * The search string used to query for users or groups



The userProviderURL and groupProviderURL strings will be used to authenticate to an LDAP server running on "localhost" and will compare any password given by the user to the "user-Password" field shown in the browser above. This will give you a .java.login.config file that looks like the following:

```
DBLogin
{
  com.sun.security.auth.module.JndiLoginModule
  required
  user.provider.url="ldap://localhost:389/ou=People,dc=BART-JENKINS-POWERBOOK-G4-17.LOCAL"
  group.provider.url="ldap://localhost:389/ou=Groups,dc=BART-JENKINS-POWERBOOK-G4-17.LOCAL";
};
```

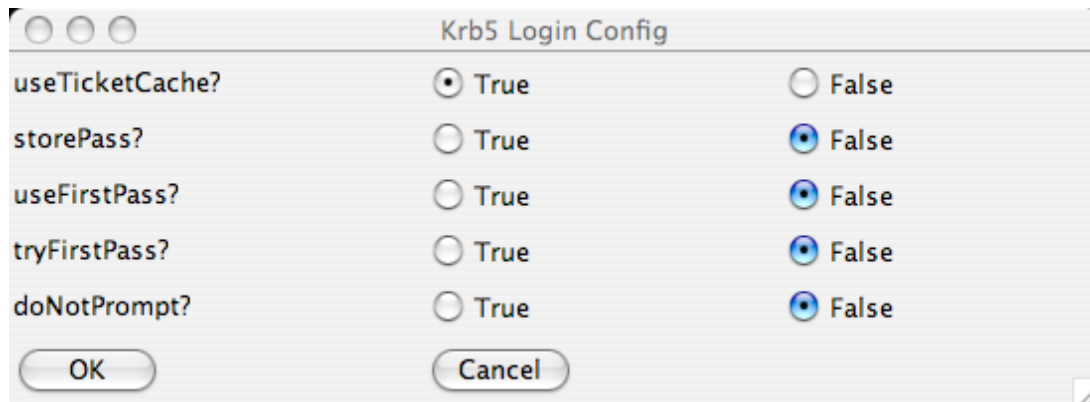
Curiously enough, the LDAP principal entry is the same as that for a Unix principal, that is:

```
principal com.sun.security.auth.UnixPrincipal "bjenkins"
```

Kerberos V

Originally developed at MIT and used (in a modified form) in Microsoft's Active Directory services in Windows 2002 and 2003 server operating systems, [Kerberos](#) is the ultimate in secure, single-signon authentication systems. AfterthoughtSoft-Secure uses the [KerberosLoginModule](#) from Sun.

The Kerberos V preferences dialog are shown below:



This will output the following for your **.java.login.config**:

```
DBLogin
{
  com.sun.security.auth.module.Krb5LoginModule required useTicketCache="true" ;
};
```

and assign a Kerberos principal in your `.java.policy` security policy file:

```
principal javax.security.auth.kerberos.KerberosPrincipal "bjenkins"
```

Then, once you have authenticated at the beginning of your day using “kinit” and been granted a Kerberos ticket, any Kerberized application will authenticate automatically, including your newly modified Java application!