**LDAP快速入门**

**1. LDAP简介**

　　LDAP（轻量级目录访问协议，Lightweight Directory Access Protocol)是实现提供被称为目录服务的信息服务。目录服务是一种特殊的数据库系统，其专门针对读取，浏览和搜索操作进行了特定的优化。目录一般用来包含描述性的，基于属性的信息并支持精细复杂的过滤能力。目录一般不支持通用数据库针对大量更新操作操作需要的复杂的事务管理或回卷策略。而目录服务的更新则一般都非常简单。这种目录可以存储包括个人信息、web链结、jpeg图像等各种信息。为了访问存储在目录中的信息，就需要使用运行在TCP/IP 之上的访问协议—LDAP。

　　LDAP目录中的信息是是按照树型结构组织，具体信息存储在条目(entry)的数据结构中。条目相当于关系数据库中表的记录；条目是具有区别名DN （Distinguished Name）的属性（Attribute），DN是用来引用条目的，DN相当于关系数据库表中的关键字（Primary Key）。属性由类型（Type）和一个或多个值（Values）组成，相当于关系数据库中的字段（Field）由字段名和数据类型组成，只是为了方便检索的需要，LDAP中的Type可以有多个Value，而不是关系数据库中为降低数据的冗余性要求实现的各个域必须是不相关的。LDAP中条目的组织一般按照地理位置和组织关系进行组织，非常的直观。LDAP把数据存放在文件中，为提高效率可以使用基于索引的文件数据库，而不是关系数据库。类型的一个例子就是mail，其值将是一个电子邮件地址。

LDAP的信息是以树型结构存储的，在树根一般定义国家(c=CN)或域名(dc=com)，在其下则往往定义一个或多个组织 (organization)(o=Acme)或组织单元(organizational units) (ou=People)。一个组织单元可能包含诸如所有雇员、大楼内的所有打印机等信息。此外，LDAP支持对条目能够和必须支持哪些属性进行控制，这是有一个特殊的称为对象类别(objectClass)的属性来实现的。该属性的值决定了该条目必须遵循的一些规则，其规定了该条目能够及至少应该包含哪些属性。例如：inetorgPerson对象类需要支持sn(surname)和cn(common name)属性，但也可以包含可选的如邮件，电话号码等属性。

**2. LDAP简称对应**

1. o– organization（组织-公司）
2. ou – organization unit（组织单元-部门）
3. c - countryName（国家）
4. dc - domainComponent（域名）
5. sn – suer name（真实名称）
6. cn - common name（常用名称）

**3. 目录设计**

设计目录结构是LDAP最重要的方面之一。下面我们将通过一个简单的例子来说明如何设计合理的目录结构。该例子将通过Netscape地址薄来访文。假设有一个位于美国US(c=US)而且跨越多个州的名为Acme(o=Acme)的公司。Acme希望为所有的雇员实现一个小型的地址薄服务器。

　　我们从一个简单的组织DN开始：

　　　　dn: o=Acme, c=US

　　Acme所有的组织分类和属性将存储在该DN之下，这个DN在该存储在该服务器的目录是唯一的。Acme希望将其雇员的信息分为两类：管理者(ou= Managers)和普通雇员(ou=Employees),这种分类产生的相对区别名(RDN,relative distinguished names。表示相对于顶点DN)就shi ：

　　　　dn: ou=Managers, o=Acme, c=US

　　　　dn: ou=Employees, o=Acme, c=US

　　在下面我们将会看到分层结构的组成：顶点是US的Acme，下面是管理者组织单元和雇员组织单元。因此包括Managers和Employees的DN组成为：

　　　　dn: cn=Jason H. Smith, ou=Managers, o=Acme, c=US

　　　　dn: cn=Ray D. Jones, ou=Employees, o=Acme, c=US

　　　　dn: cn=Eric S. Woods, ou=Employees, o=Acme, c=US

　　为了引用Jason H. Smith的通用名(common name )条目，LDAP将采用cn=Jason H. Smith的RDN。然后将前面的父条目结合在一起就形成如下的树型结构：

　　　　cn=Jason H. Smith

　　　　　　　　+ ou=Managers

　　　　　　　　　　　　+ o=Acme

　　　　　　　　　　　　　　　　+ c=US

　                              -> dn: cn=Jason H. Smith,ou=Managers,o=Acme,c=US

　　现在已经定义好了目录结构，下一步就需要导入目录信息数据。目录信息数据将被存放在LDIF文件中，其是导入目录信息数据的默认存放文件。用户可以方便的编写Perl脚本来从例如/etc/passwd、NIS等系统文件中自动创建LDIF文件。

　　下面的实例保存目录信息数据为testdate.ldif文件，该文件的格式说明将可以在man ldif中得到。

　　在添加任何组织单元以前，必须首先定义Acme DN：

　　　　dn: o=Acme, c=US

　　　　objectClass: organization

　　这里o属性是必须的

　　　　o: Acme

　　下面是管理组单元的DN，在添加任何管理者信息以前，必须先定义该条目。

　　　　dn: ou=Managers, o=Acme, c=US

　　　　objectClass: organizationalUnit

这里ou属性是必须的。

ou: Managers

　　第一个管理者DN：

　　　　dn: cn=Jason H. Smith, ou=Managers, o=Acme, c=US

　　　　objectClass: inetOrgPerson

　　cn和sn都是必须的属性：

　　　　cn: Jason H. Smith

　　　　sn: Smith

　　但是还可以定义一些可选的属性：

　　　　telephoneNumber: 111-222-9999

　　　　mail: headhauncho@acme.com

　　　　localityName: Houston

　　可以定义另外一个组织单元：

　　　　dn: ou=Employees, o=Acme, c=US

　　　　objectClass: organizationalUnit

　　　　ou: Employees

　　并添加雇员信息如下：

　　　　dn: cn=Ray D. Jones, ou=Employees, o=Acme, c=US

　　　　objectClass: inetOrgPerson

　　　　cn: Ray D. Jones

　　　　sn: Jones

　　　　telephoneNumber: 444-555-6767

　　　　mail: jonesrd@acme.com

　　　　localityName: Houston

　　　　dn: cn=Eric S. Woods, ou=Employees, o=Acme, c=US

　　　　objectClass: inetOrgPerson

　　　　cn: Eric S. Woods

　　　　sn: Woods

　　　　telephoneNumber: 444-555-6768

　　　　mail: woodses@acme.com

　　　　localityName: Houston

**4. 配置OpenLDAP**

本文实践了在 Windows 下安装配 openldap，并添加一个条目，LdapBrowser 浏览，及 Java 程序连接 openldap 的全过程。

1. 下载安装 **openldap** for windows，当前版本2.2.29下载地址：http://download.bergmans.us/openldap/openldap-2.2.29/openldap-2.2.29-db-4.3.29-openssl-0.9.8a-win32\_Setup.exe

    相关链接：http://lucas.bergmans.us/hacks/openldap/

   安装很简单，一路 next 即可，假设我们安装在 c:\openldap

2. 配置 openldap，编辑 sldap.conf 文件

   1) 打开 c:\openldap\sldap.conf，找到

    include  C:/openldap/etc/schema/core.schema，在它后面添加

    include  C:/openldap/etc/schema/cosine.schema

    include  C:/openldap/etc/schema/inetorgperson.schema

    接下来的例子只需要用到以上三个 schema，当然，如果你觉得需要的话，你可以把其他的 schema 全部添加进来

    include  C:/openldap/etc/schema/corba.schema

    include  C:/openldap/etc/schema/dyngroup.schema

    include  C:/openldap/etc/schema/java.schema

    include  C:/openldap/etc/schema/misc.schema

    include  C:/openldap/etc/schema/nis.schema

    include  C:/openldap/etc/schema/openldap.schema

   2) 还是在 sldap.conf 文件中，找到

    suffix  "dc=my-domain,dc=com"

    rootdn  "cn=Manager,dc=my-domain,dc=com"

    把这两行改为

    suffix "o=teemlink,c=cn"

    rootdn "cn=Manager,o=teemlink,dc=cn"

    suffix 就是看自己如何定义了，后面步骤的 ldif 文件就必须与它定义了。还要注意到这个配置文件中有一个 rootpw  secret，这个 secret 是 cn=Manager 的密码，以后会用到，不过这里是明文密码，你可以用命令： slappasswd -h {MD5} -s secret 算出加密的密码 {MD5}Xr4ilOzQ4PCOq3aQ0qbuaQ== 取代配置中的 secret。

3. 启动 openldap

    CMD 进入到 c:\openldap 下，运行命令 sldapd -d 1

    用可以看到控制台下打印一片信息，openldap 默认是用的 Berkeley DB 数据库存储目录数据的。

4. 建立条目,编辑导入 ldif 文件

   1) 新建一个 ldif(LDAP Data Interchanged Format) 文件(纯文本格式)，例如 test.ldif，文件内容如下：

dn: o=teemlink

objectclass: top

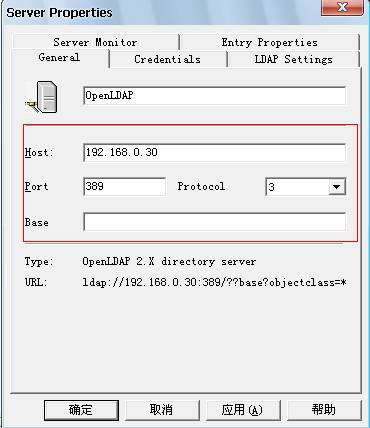
objectclass: organization

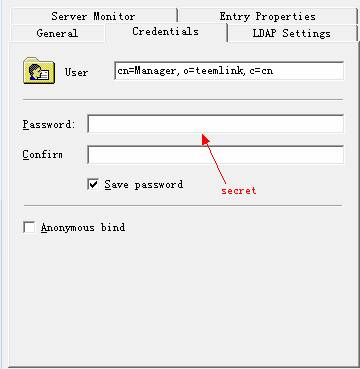
o: develop

   2) 执行命令：ldapadd -l test.ldif

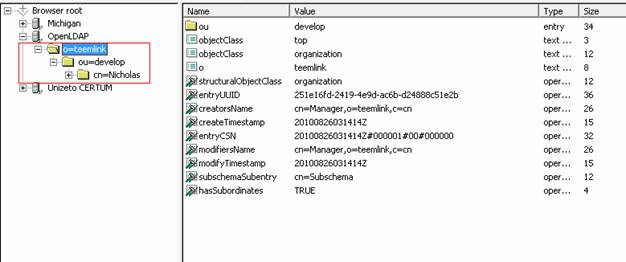
5. 使用LDAP Browser进行访问

       5.1安装LDAP Browser2.6软件，进行如下操作：





5.2显示效果



**5. Java操作LDAP**

**5.1 用JNDI进访问**

[复制代码](javascript:void(0);)

package cn.myapps.test;  
  
import java.util.Hashtable;  
import javax.naming.Context;  
import javax.naming.NamingException;  
import javax.naming.directory.Attributes;  
import javax.naming.directory.DirContext;  
import javax.naming.directory.InitialDirContext;  
  
public class LdapTest {  
 public void JNDILookup() {  
 String root = "o=teemlink,c=cn";  
 Hashtable env = new Hashtable();  
 env.put(Context.INITIAL\_CONTEXT\_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");  
 env.put(Context.PROVIDER\_URL, "ldap://192.168.0.30/" + root);  
 env.put(Context.SECURITY\_AUTHENTICATION, "simple");  
 env.put(Context.SECURITY\_PRINCIPAL, "cn=Nicholas,ou=产品,o=teemlink,c=cn");  
 env.put(Context.SECURITY\_CREDENTIALS, "123456");  
 DirContext ctx = null;  
  
 try {  
 ctx = new InitialDirContext(env);  
 Attributes attrs = ctx.getAttributes("cn=Nicholas,ou=产品");  
 System.out.println("Last Name: " + attrs.get("sn").get());  
 System.out.println("认证成功");  
 } catch (javax.naming.AuthenticationException e) {  
 e.printStackTrace();  
 System.out.println("认证失败");  
 } catch (Exception e) {  
 System.out.println("认证出错：");  
 e.printStackTrace();  
 }  
 if (ctx != null) {  
 try {  
 ctx.close();  
 } catch (NamingException e) {  
 // ignore  
 }  
 }  
 }  
  
 public static void main(String[] args) {  
 LdapTest LDAPTest = new LdapTest();  
 LDAPTest.JNDILookup();  
 }  
}

[复制代码](javascript:void(0);)

**5.2 用JLDAP进访问**

访问地址：<http://www.openldap.org/jldap/> 并下载相关lib

[复制代码](javascript:void(0);)

import com.novell.ldap.\*;  
  
import java.io.UnsupportedEncodingException;  
  
public class List  
  
{  
  
 public static void main(String[] args)  
  
 {  
 int ldapPort = LDAPConnection.DEFAULT\_PORT;  
 int searchScope = LDAPConnection.SCOPE\_ONE;  
 int ldapVersion = LDAPConnection.LDAP\_V3;  
 boolean attributeOnly = false;  
 String attrs[] = null;  
 String ldapHost = "192.168.0.30";  
 String loginDN = "cn=Manager,o=teemlink,c=cn";  
 String password = "secret";  
 String searchBase = "ou=develop,o=teemlink,c=cn";  
 String searchFilter = "objectClass=\*";  
  
 LDAPConnection lc = new LDAPConnection();  
 try {  
 // connect to the server  
 lc.connect(ldapHost, ldapPort);  
  
 // bind to the server  
 lc.bind(ldapVersion, loginDN, password.getBytes("UTF8"));  
  
 LDAPSearchResults searchResults =  
  
 lc.search(searchBase, // container to search  
 searchScope, // search scope  
 searchFilter, // search filter  
 attrs, // "1.1" returns entry name only  
 attributeOnly); // no attributes are returned  
  
 // print out all the objects  
 while (searchResults.hasMore()) {  
 LDAPEntry nextEntry = null;  
 try {  
 nextEntry = searchResults.next();  
 System.out.println("\n" + nextEntry.getDN());  
 System.out.println(nextEntry.getAttributeSet());  
 } catch (LDAPException e) {  
 System.out.println("Error: " + e.toString());  
 // Exception is thrown, go for next entry  
 continue;  
 }  
 }  
  
 // disconnect with the server  
 lc.disconnect();  
  
 } catch (LDAPException e) {  
 System.out.println("Error: " + e.toString());  
 } catch (UnsupportedEncodingException e) {  
 System.out.println("Error: " + e.toString());  
 }  
 System.exit(0);  
 }  
}

[复制代码](javascript:void(0);)

**5.3 用JDBC-LDAP进访问**

访问地址：<http://www.openldap.org/jdbcldap/> 并下载相关lib

[复制代码](javascript:void(0);)

package jdbcldap;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
  
public class JdbcLdap {  
  
 /\*\*  
 \* @param args  
 \* @throws Exception   
 \*/  
 public static void main(String[] args) throws Exception {  
 Class.forName("com.octetstring.jdbcLdap.sql.JdbcLdapDriver");  
 String ldapConnectString = "jdbc:ldap://192.168.0.30/o=teemlink,c=cn?SEARCH\_SCOPE:=subTreeScope";  
 Connection con = DriverManager.getConnection(ldapConnectString, "cn=Manager,o=teemlink,c=cn", "secret");  
  
 String sql = "SELECT \* FROM ou=develop,o=teemlink,c=cn";  
  
 Statement sat = con.createStatement();  
 ResultSet rs = sta.executeQuery(sql);  
 while (rs.next()) {  
 System.out.println(rs.getString(1));  
 }  
  
 if (con != null)  
 con.close();  
 }  
}