

公告

随笔-32 文章-0 评论-450

C# AD(Active Directory)域信息同步，组织单位、用户等信息查询

接上篇 [Windows Server 2008 R2 配置AD\(Active Directory\)域控制器](#)

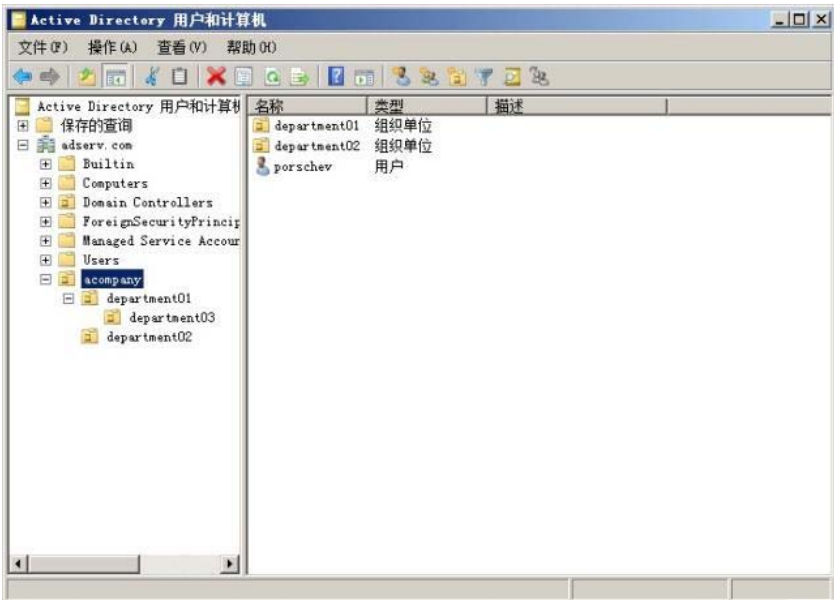
对AD域结合常见需求用C#进行一些读取信息的操作^_^!

目录

- [示例准备](#)
- [知识了解](#)
- [读取AD域信息示例](#)
- [DirectorySearcher.Filter属性扩充说明](#)
- [用户属性扩充说明\(含图文属性对照\)](#)
 - [常规](#)
 - [地址](#)
 - [帐户](#)
 - [电话](#)
 - [组织](#)
- [示例下载](#)

示例准备

- 打开 [上一篇文章](#)配置好的AD域控制器
- 开始菜单-->管理工具-->Active Directory 用户和计算机
- 新建组织单位和用户



- 新建层次关系如下：



[回到顶部](#)

知识了解

我们要用C#访问Active Directory非常容易，主要用到

[轻量目录访问协议 \(LDAP\)](#)

[System.DirectoryServices](#)命名空间下的两个组件类

[DirectoryEntry](#)和[DirectorySeacher](#)

读取AD域信息示例

示例在Framework 3.5下用Winform程序编写

主要结合常见需求读取组织单位(OU)及用户(User)信息，以及同步组织单位和用户的层次关系;

比较着重的还是用户的信息，特别是帐号、邮箱、SID等信息;

- 下面我们开始连接域，并读取出**示例准备**中键好的组织单位和用户

首先编写代码用LDAP尝试对域进行访问

形式: LDAP://Domain

```
#region## 是否连接到域
/// <summary>
/// 功能: 是否连接到域
/// 作者: Wilson
/// 时间: 2012-12-15
/// http://msdn.microsoft.com/zh-cn/library
/system.directoryservices.directoryentry.path(v=vs.90).aspx
/// </summary>
/// <param name="domainName">域名或IP</param>
/// <param name="userName">用户名</param>
/// <param name="userPwd">密码</param>
/// <param name="entry">域</param>
/// <returns></returns>
private bool IsConnected(string domainName, string userName, string userPwd, out
DirectoryEntry domain)
{
    domain = new DirectoryEntry();
    try
    {
        domain.Path = string.Format("LDAP://{0}", domainName);
        domain.Username = userName;
        domain.Password = userPwd;
        domain.AuthenticationType = AuthenticationTypes.Secure;

        domain.RefreshCache();

        return true;
    }
    catch(Exception ex)
    {
        LogRecord.WriteLog("[IsConnected方法]错误信息: " + ex.Message);
        return false;
    }
}
#endregion
```

传用参数，调IsConnected方法，结果如下

[回到顶部](#)



- 连接上AD域后，接着我们找到根OU

```
#region## 域中是否存在组织单位
/// <summary>
/// 功能：域中是否存在组织单位
/// 作者：Wilson
/// 时间：2012-12-15
/// </summary>
/// <param name="entry"></param>
/// <param name="ou"></param>
/// <returns></returns>
private bool IsExistOU(DirectoryEntry entry, out DirectoryEntry ou)
{
    ou = new DirectoryEntry();
    try
    {
        {
            ou = entry.Children.Find("OU=" + txtRootOU.Text.Trim());

            return (ou != null);
        }
    }
    catch(Exception ex)
    {
        LogRecord.WriteLog("[IsExistOU方法]错误信息: " + ex.Message);
        return false;
    }
}
#endregion
```

传入以数，调用IsExistOU方法，结果如下



- 下面开始读取组织单位及用户的信息。

示例为了看出层次关系及导出信息是类型区分，给OU和User新建了一个实体类和一个类型的：

```
#region## 类型
/// <summary>
/// 类型
/// </summary>
public enum TypeEnum : int
{
    /// <summary>
```

[回到顶部](#)

```
/// 组织单位
/// </summary>
OU = 1,

/// <summary>
/// 用户
/// </summary>
USER = 2
}
#endregion

#region## Ad域信息实体
/// <summary>
/// Ad域信息实体
/// </summary>
public class AdModel
{
    public AdModel(string id, string name, int typeId, string parentId)
    {
        Id = id;
        Name = name;
       TypeId = typeId;
        ParentId = parentId;
    }

    public string Id { get; set; }

    public string Name { get; set; }

    public int TypeId { get; set; }

    public string ParentId { get; set; }
}
#endregion
```

下面读取信息

```
private List<AdModel> list = new List<AdModel>();

#region## 同步
/// <summary>
/// 功能:同步
/// 创建人:Wilson
/// 创建时间:2012-12-15
/// </summary>
/// <param name="entryOU"></param>
public void SyncAll(DirectoryEntry entryOU)
{
    DirectorySearcher mySearcher = new DirectorySearcher(entryOU,
"(objectclass=organizationalUnit)"); //查询组织单位

    DirectoryEntry root = mySearcher.SearchRoot;    //查找根OU

    SyncRootOU(root);

    StringBuilder sb = new StringBuilder();

    sb.Append("\r\nID\帐号\t类型\t父ID\r\n");

    foreach (var item in list)
    {
        sb.AppendFormat("{0}\t{1}\t{2}\t{3}\r\n", item.Id, item.Name,
item.TypeId, item.ParentId);
    }

    LogRecord.WriteLog(sb.ToString());

    MessageBox.Show("同步成功", this.Text, MessageBoxButtons.OK,
MessageBoxIcon.Information);

    Application.Exit();
}
#endregion
```

[回到顶部](#)

```
#region## 同步根组织单位
/// <summary>
/// 功能: 同步根组织单位
/// 创建人:Wilson
/// 创建时间:2012-12-15
/// </summary>
/// <param name="entry"></param>
private void SyncRootOU(DirectoryEntry entry)
{
    if (entry.Properties.Contains("ou") &&
entry.Properties.Contains("objectGUID"))
    {
        string rootOuName = entry.Properties["ou"][0].ToString();

        byte[] bGUID = entry.Properties["objectGUID"][0] as byte[];

        string id = BitConverter.ToString(bGUID);

        list.Add(new AdModel(id, rootOuName, (int)TypeEnum.OU, "0"));

        SyncSubOU(entry, id);
    }
}
#endregion

#region## 同步下属组织单位及下属用户
/// <summary>
/// 功能: 同步下属组织单位及下属用户
/// 创建人:Wilson
/// 创建时间:2012-12-15
/// </summary>
/// <param name="entry"></param>
/// <param name="parentId"></param>
private void SyncSubOU(DirectoryEntry entry, string parentId)
{
    foreach (DirectoryEntry subEntry in entry.Children)
    {
        string entrySchemaClsName = subEntry.SchemaClassName;

        string[] arr = subEntry.Name.Split('=');
        string categoryStr = arr[0];
        string nameStr = arr[1];
        string id = string.Empty;

        if (subEntry.Properties.Contains("objectGUID")) //SID
        {
            byte[] bGUID = subEntry.Properties["objectGUID"][0] as byte[];

            id = BitConverter.ToString(bGUID);
        }

        bool isExist = list.Exists(d => d.Id == id);

        switch (entrySchemaClsName)
        {
            case "organizationalUnit":

                if (!isExist)
                {
                    list.Add(new AdModel(id, nameStr, (int)TypeEnum.OU,
parentId));
                }

                SyncSubOU(subEntry, id);
                break;
            case "user":
                string accountName = string.Empty;

                if (subEntry.Properties.Contains("samaccountName"))
                {
                    accountName = subEntry.Properties["samaccountName"]
[0].ToString();
                }
            }
        }
    }
}
```

[回到顶部](#)

```
        }

        if (!isExist)
        {
            list.Add(new AdModel(id, accountName, (int)TypeEnum.USER,
parentId));
        }
        break;
    }
}
}
#endregion
```

调用SyncAll方法循环输出list,结果如下,很清楚的可以看出层次关系

ID	帐号	类型	父
//58-D6-C4-32-6A-A1-99-48-A4-8B-C8-5D-BC-C9-3E-17	acompany	1	0
//FB-44-91-AE-AC-73-2B-4D-9F-01-B1-E2-16-D3-CB-1B	department01	1	
58-D6-C4-32-6A-A1-99-48-A4-8B-C8-5D-BC-C9-3E-17			
//47-9D-5B-91-60-22-D1-46-B0-CD-C7-B2-C7-D3-00-31	department03	1	
FB-44-91-AE-AC-73-2B-4D-9F-01-B1-E2-16-D3-CB-1B			
//E3-AD-47-45-38-64-02-4D-B9-83-2C-50-67-50-4F-92	zw	2	
47-9D-5B-91-60-22-D1-46-B0-CD-C7-B2-C7-D3-00-31			
//8A-D4-23-18-F3-6F-E1-47-93-7A-CC-07-76-4B-E7-86	zhongw	2	
FB-44-91-AE-AC-73-2B-4D-9F-01-B1-E2-16-D3-CB-1B			
//BC-D0-34-85-67-2F-05-4D-B5-77-E3-F4-AD-51-45-02	department02	1	
58-D6-C4-32-6A-A1-99-48-A4-8B-C8-5D-BC-C9-3E-17			
//1C-13-FA-66-E4-51-65-49-8B-DC-22-60-32-34-8F-22	wilson	2	
BC-D0-34-85-67-2F-05-4D-B5-77-E3-F4-AD-51-45-02			
//84-E8-E5-9A-6B-56-E2-45-9A-87-54-D1-78-6B-D3-56	porschev	2	
58-D6-C4-32-6A-A1-99-48-A4-8B-C8-5D-BC-C9-3E-17			

DirectorySearcher.Filter属性扩充说明

```
DirectorySearcher mySearcher = new DirectorySearcher(entryOU,
"(objectclass=organizationalUnit)"); //查询组织单位
```

第二个参数是一个filter，也可以根据需求输入其它筛选条件，下面列出几个常用的

筛选条件	值
用户	(&(objectCategory=person)(objectClass=user))
计算机	(objectCategory=computer)
组	(objectCategory=group)
联系人	(objectCategory=contact)
共享文件夹	(objectCategory=volume)
打印机	(objectCategory=printQueue)

更多高级筛选请查看: [http://msdn.microsoft.com/zh-cn/library/system.directoryservices.directorysearcher.filter\(v=vs.80\).aspx](http://msdn.microsoft.com/zh-cn/library/system.directoryservices.directorysearcher.filter(v=vs.80).aspx)

用户属性扩充说明(含图文属性对照)

示例中只对用户进行了读取了几个属性，用过AD域的应该都知道，用户的属性较多也比较常
下面通过AD域的用户详细信来对照一下相应的属性名

[回到顶部](#)

- 常项选项卡

perschev 属性

拨入 环境 会话 远程控制
远程桌面服务配置文件 个人虚拟机 COM+

常规 地址 帐户 配置文件 电话 组织 隶属于

perschev

姓(L): ① perschev
名(F): ② 英文缩写(I): ⑥
显示名称(S): ③ perschev
描述(D): ④
办公室(C): ⑤
电话号码(T): ⑦ 其他(O): ⑩
电子邮件(M): ⑧
网页(W): ⑨ 其他(R): ⑪

确定 取消 应用(A) 帮助

对应编号	选项卡对应项名	属性名
①	姓(L)	sn
②	名(F)	givenName
③	显示名称(S)	displayName
④	描述(D)	description
⑤	办公室(C)	physicalDeliveryOfficeName
⑥	英文缩写(I)	initials
⑦	电话号码(T)	telephoneNumber
⑧	电子邮件(M)	mail
⑨	网页(W)	wwwHomePage
⑩	电话号码-其它(O)...	otherTelephone
⑪	网页-其它(R)...	url

- 地址选项卡

perschev 属性

拨入 环境 会话 远程控制
远程桌面服务配置文件 个人虚拟机 COM+

常规 地址 帐户 配置文件 电话 组织 隶属于

国家/地区(O): ①
省/自治区(V): ②
市/县(C): ③
街道(S): ④
邮政信箱(B): ⑤
邮政编码(Z): ⑥

确定 取消 应用(A) 帮助

对应编号	选项卡对应项名	属性名
①	国家/地区(O)	co
②	省/自治区(V)	st
③	市/县(C)	l

[回到顶部](#)

④	街道(S)	streetAddress
⑤	邮政信箱(B)	postOfficeBox
⑥	邮政编码(Z)	postalCode

• 帐户选项卡

帐户选项卡内容：

- 用户登录名(U): porschev @ad.com ①
- 用户登录名(Windows 2000 以前版本)(W): AD\porschev ②
- 登录时间(L)... 登录到(T)...
- ☐ 解锁帐户(U)
- 帐户选项(O):
 - ☐ 用户下次登录时须更改密码
 - ☐ 用户不能更改密码
 - ☒ 密码永不过期
 - ☒ 使用可逆加密存储密码
- 帐户过期:
 - ☒ 永不过期(Y)
 - ☐ 在这之后(Z): 2013年 1月13日

对应编号	选项卡对应项名	属性名
①	用户登录名(U)	userPrincipalName
②	用户登录名(Windows 2000 以前版本)(W)	sAMAccountName

• 电话选项卡

电话选项卡内容：

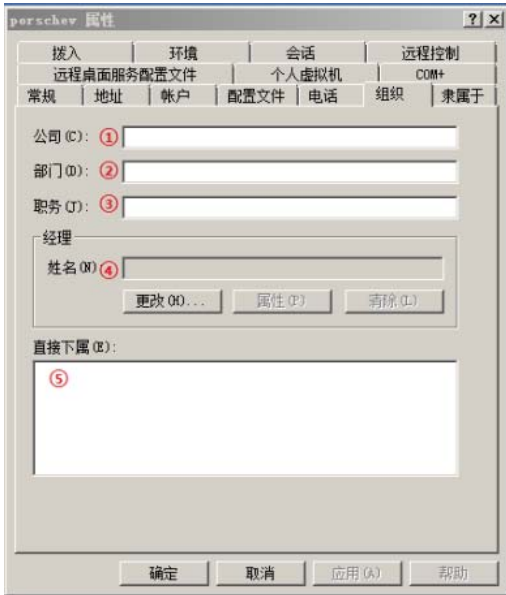
- 电话号码
- 家庭电话(M): ① 其他(O): ⑦
- 寻呼机(P): ② 其他(O): ⑧
- 移动电话(B): ③ 其他(O): ⑨
- 传真(F): ④ 其他(O): ⑩
- IP 电话(I): ⑤ 其他(O): ⑪
- 注释(N): ⑥

对应编号	选项卡对应项名	属性名
①	家庭电话(M)	homePhone
②	寻呼机(P)	pager
③	移动电话(B)	mobile
④	传真(F)	facsimileTelephoneNumber
⑤	IP电话(I)	ipPhone
⑥	注释	info

[回到顶部](#)

⑦	家庭电话-其它(O)	otherHomePhone
⑧	寻呼机-其它(T)	otherPager
⑨	移动电话-其它(B)	otherMobile
⑩	传真-其它(E)	otherFacsimileTelephoneNumber
⑪	IP电话-其它(R)	otherIpPhone

• 组织选项卡



对应编号	选项卡对应项名	属性名
①	公司(C)	company
②	部门(D)	department
③	职务(J)	title
④	经理-姓名(N)	manager
⑥	直接下属(E)	directReports

还有一些属性没有列出来，可以循环输出DirectoryEntry.Properties.PropertyNames来找
比如用objectsid这也是个用户比较重要的属性，在设置Windows共享时会用到!

示例下载

示例下载: <http://files.cnblogs.com/zhongweiv/SynchronousAD.zip>
示例代码，写得比较简陋，有需要就下载了将就看一下吧^_^!

posted @ 2013-01-05 14:05 porschev 阅读(2608) 评论(30) !

[刷新评论](#) [刷新页面](#) [返回顶部](#)

[博客园首页](#) [博问](#) [新闻](#) [闪存](#) [程序员招聘](#) [知识库](#)