Windows 2003 Server 活动目录研究

邓文亮, 倪子伟
(厦门大学计算机系 福建 厦门 361012)

【摘 要】: 目前, 大部分大中型企业都部署了 Windows 2003 Server 和 MS Exchange Server, 管理它们就成为 IT 部门的一项重要工作。本文提出了如何用 ADSI 来管理企业内部的账号和邮箱, 以便于利用 ASP 技术, 通过 Web 方式来方便其管理。

【关键字】: windows 2003 Server、MS Exchange Server、ADSI、ASP、Web

1. 引 言

目前大部分企业是通过 Windows server 活动目录提供的功能来管理其内部的计算机和用户账户, 并且相应建立起了邮件服务器, 为每个账户分配邮箱, 方便员工相互通讯, 降低企业的运营成本。

账户数量较少时管理比较容易, 但是随着账户数量的增加, 在管理方面出现很多问题。比如重新安装 Active Directory 时, 原有的账户数据就会全部丢失。如果可以通过 Web 方式对账户进行增、删、改、查和数据额外备份的话, 将会解决数据丢失这一问题, 使管理变得更方便。如此, 我们的目的就是让企业账户的管理更加方便, 以便实现账户数据的备份和远程管理 AD。

2. 概 述

2.1 Active Directory

活动目录 (Active Directory, AD)不仅是一个管理工具, 可以管理整个企业的计算机、打印机、用户账户等; 同时也是用户工具, 使用户能够轻易地得到所需资源为之服务; 再者, 它与 Microsoft Exchange Server 的融合, 为企业内部的通讯提供了巨大的便利。

2.2 LDAP

轻量级目录访问协议 (Light Directory Access Protocol, LDAP)是一种目录服务协议, 对 AD 的资源信息规范化。LDAP 对信息的管理规范化, 主要表现在以下两个方面:

1. 层次化(或者是树型 DIT)命名模型

通俗地说是 LDAP 中的条目定位方式, 可区别的名称(distinguishedName, DN), 是有相关可区别名称 (Relative distinguishedName, RDN) 以及 RDN 所在的容器的 DN 构成。如图 1, Jessy 的 RDN 是 CN=Jessy, 完整的 DN 是 CN =Jessy,OU =MIS, DC=Microsoft,DC=COM。

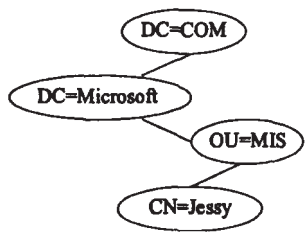

图 1

2. LDAP 的实现简单化

LDAP 继承了 X.500 最好的特性, 同时去掉了它的复杂性。LDAP 通过使用查找操作实现列表操作和读操作, 另一方面省去了 X.500 中深奥的和很少使用的服务控制和安全特性, 只保留常用的特性, 简化了 LDAP 的实现。

2.3 ADSI 编程

活动目录服务接口 (Active Directory Service Interface, ADSI)是 Microsoft 公司提供的脚本接口, 是 AD 应用程序编程接口 (Application Program Interface, API)。可以通过 ADSI 编程简单地实现对 LDAP 名字空间进行访问。ADSI 对象模型由 COM(Component Object Model, 组建对象模型)对象组成, 支持目录查询和对象管理。同时, ADSI 也是 OLE DB 的提供商, 所以任何已经使用 OLE DB 的客户, 包括使用 ADO 的客户, 都可以对 AD 进行查询操作。

2.3.1 ADSI 绑定串格式

支持 LDAP 的编程语言普遍具有以下格式:

LDAP://HOSTNAME[:portnumber]/DN

其中, HOSTNAME 可以是一个主机名, 一个 IP 地址或者域名, DN 是 AD 对象的区别名。如: LDAP://VirtulPC/ CN=Jessy, OU=MIS, DC=Mcrosoft, DC=COM, VirtualPC 是主机名, CN=Jessy, OU=MIS, DC=Mcrosoft, DC=COM 是用户账户 Jessy 的区别名。

2.3.2 ADSI 编程绑定对象

组织(organization)、组(gorup)和用户(user)等是 AD 中的对象。通过 ADSI 来操 AD 的对象和数据库, 这就是 ADSI 提供的绑定对象的两种方式:

1. 用 ADSI 函数绑定到一个对象, 主要有两个函数:

• ADsGetObject : 用当前凭证绑定到一个 ADSI 对象。GetObject 函数有一个参数, 这个参数指明了要遵循的协议以及在该协议下对象的 DN。这样就可以获取一个 COM 标记(moniker), 这个标记指向目录对象。

绑定域控制器: Set objDm = GetObject(LDAP://DC=Mcrosoft, DC=COM)

绑定组织: Set objOu = GetObject (LDAP:// OU=MIS, DC=Mcrosoft, DC=COM)

绑定组织: Set objGroup = GetObject (LDAP:// CN=ErpTeam, OU=MIS, DC=Mcrosoft, DC=COM)

绑定用户: Set objUser = GetObject (LDAP://CN=Jessy,OU=MIS, DC=Mcrosoft, DC=COM)

• ADsOpenObject: 该方法适用于需要建立信任凭证, 确保安全, 所以需要指定用户名和口令, 该用法的详细介绍请参考相关文献。

对象绑定后, 就返回一个具体的实例, 如 objOu、objGroup, 通过实例可以调用相关的方法和提取其各属性值。

2. 通过 ADO 绑定数据对象

因为 ADSI 也是一个 OLE DB 提供商, 所以我们可以将 ADO 连接到活动目录数据库, 如程式 2- 3。

```
1. Function getCon()
2. Set Con = CreateObject("ADODB.Connection")
3. Con.Provider = "ADsDSObject"
4. Con.Open "Active Directory"
5. getCon = Con
6. End Function
```

程式 2- 1 连接 AD 数据库

程式 2- 1 是以当前用户的身份连接数据库, 我们也可以使用连接特性来指定不同的信任凭证, 请参考相关文献。数据库连接成功后, 我们就可以通过 SQL 语句来绑定到指定的对象, 示例代码如下:

```
1. Function srchOu(ouName)
2. Set Con = getCon()
3. Set domain = getDm()  ' 获取域的名称, 如程式 2- 3
4. sqlStr = "SELECT distinguishedName form 'LDAP://"& domain &" where objectClass = 'organizationalUnit' and ou='" & ouName & "'"
5. Set Rs = Con.Execute(sqlStr)
6. srchOu = Rs("distinguishedName")
7.      ……
8.      End Function
```

程式 2- 2 通过 ADO 绑定组织

通过 ADO 绑定数据对象, 可以得到所需要的对象的属性

值。程式 2-2 演示了如何用 ADO 绑定到组织，函数的返回值为组织的 DN。当然，我们可以在 SELECT 子句中添加其它属性字段得到对应的值，也可以在 WHERE 子句中添加或修改约束条件来绑定不同的对象或同一个对象的不同实例。比如，做如下修改后可以绑定到组和用户：

将 WHERE 子句中的属性 objectClass 的值改成"User"可以绑定到用户，当然，函数名也相应改成 srchUser(usrName): sqlStr = "SELECT distinguishedName form 'LDAP://"& domain &" where objectClass = 'User' and name='" & usrName & "'"。

将 WHERE 子句中的属性 objectClass 的值改成"Group"可以绑定到组，当然，函数名也相应改成 srchGroup(grpName): sqlStr = "SELECT distinguishedName form 'LDAP://"& domain &" where objectClass = 'Group' and name='" & grpName & "'"。

III. 此外，为了便于编写程式，我们可以通过下面代码获得域的 LDAP 名称。

```
1. Function getDm()
2. Set rootDSE = GetObject("LDAP://RootDSE") '
3. Set domain = rootDSE.Get("defualtNamingContext")
4. getDm = domain
5. End Function
```

程式 2-3　获得域的 LDAP 名称

2.4 Microsoft Exchange Server

Microsoft Exchange Server 提供了与 AD 相关联的目录服务，为 AD 中的账户（组和用户）提供了邮件服务。那么，对 AD 的编程就有必要去访问 Exchange，幸好利用 ADSI 所提供的功能，可以编写客户程序访问 Microsoft Exchange Server。

3. 账户(Account)管理

对一个大型的企业来说，用 Windows Server 来管理企业的各种资源是非常重要的，其中 Active Directory 工具就提供了部分功能，但是目前直接用 Web 来管理 AD 是相对较少的，这样，专业人士对 AD 的研究就显得很重要了。

3.1 组织(Organization Unit)管理

在一个企业中，组织的定义可以按照各自的特点来定义，比如某个大型企业，他们有技术部、工程部、管理、贸易部等等，可以把他们定义为组织。组织也是一个容器，它的成员(members)包括子组织，组和用户。下面介绍对组织的操作。

I. 组织的查询操作 (srchOu (ouName)) 和绑定操作(bindOu (ouname))请参考 2.3.2 节。

II. 创建组织

程序 3-1 是如何创建一个组织。

```
1. Function OuCreate(ouName, fatherOu)
2. objFatherOu = bindOu(fatherOu)
3. Set objOu = objFatherOu.Create("organizationUnit", "ou=" &  ouName)
4. objOu.SetInfo
5. End Function
```

程式 3-1　组织创建

程式 3-1 中有两个参数：ouName 代表要创建的组织的名称；fatherOu 代表父组织的名称。第 4 行中的 setInfo 方法是将对象的特性的当前值从特性高速缓存器保存到低层的目录存储器中，使得对象创建，改变或删除生效。

III. 删除组织

组织的创建比较简单，因为对组织属性的操作比较少。相应地，组织的删除也比较简单，只需将程式 3-1 第 8 行改成：objFatherOu.Delete "organizationUnit", "ou=" & ouName 即可。当然 objFatherOu 也可能是域控制器对象，但是组织创建和删除的方法是一样的，只需要将 objFatherOu 绑定到域控制器就可以了。

IV. 列举组织的成员

组织是一个容器，它是包含子组织、用户和组的集合。所以可以通过下述方法来列举它的成员 members。显然，members 是一个集合，可以通过 For each 语句来列举。列举 members 的方法如下：

```
1. Function listMembers(ouName)
2. Set objOu = bindOu(ouName)
3. members = objOu.members
4. For each member in members
5. Each.write(member.name)
6. Next
7. End Function
```

程式 3-2　列举组织成员

如果要对 members 进行过滤，可以通过 ADSI 的内置函数 fileter 来实现：

滤出用户：UsrList = members.filter("user");

滤出组：GrpList = members.filter("group");

滤出子组织：OuList = members.filter("organizationalUnit")。

3.2 用户(User)管理

在一个企业中，用户的定义很简单，它就是系统分配的一个账号，它可以被普通用户使用，也可以被一个程式使用。查询用户（srchUser(usrName) 和绑定用户 bindUser(usrName)在上文中已谈到。

I. 创建用户

```
1. Function createUser (ouName,sn,givenName,displayName,loginName,password,company,department,userAccoutControl)
2. Set objFatherOu = bindOu(ouName)
3. Set objUser = objFatherOu.Create("User", "CN=" & loginName )
4. objUser.Put "sn",sn
5. ……  ' 设置其它属性
6. objUser.SetInfo
7. objUser.AccountDisabled = False  ' 启用账户
8.objUser.SetPassword password  ' 设置账户密码
9.objUser.Put "userAccoutControl", userAccoutControl ' 设置账户属性
10. objUser.SetInfo
11. End Function
```

程式 3-3　创建用户

从程式 3-3 可以看出用户也是通过组织对象的方法 Create 来创建的。不同点的地方是账户属性设置，如用户的密码和账户属性。此外，userAccoutControl 是用户的整型变量，是账户属性的组合值，如：其值为 66080 表示账户永不过期和用户不能修改密码。另外账户的禁用和启用有两种方法：

通过用户对象的属性 AccountDisabled：其值为 True 代表账户禁用 false 代表账户启用；

改变 userAccoutControl 值；如果原来账户禁用，将 userAccoutControl 减 2 即可启用账户；相反，则禁用账户。

II. 删除用户：用户的删除也是通过组织对象的方法 delete() 来实现的，如：objFatherOu.delete "User","CN=Jessy"。

III. 属性设置和修改：账户的属性设置和修改都可以通过 put 或 putEx 来实现，只不过修改属性是将原值覆盖或替换而已，当然，修改属性之前要绑定到对象，这样才能调用对象的方法 put 和 putEx。注意，有些属性是数组，即它们是账户的多值属性，如 dLMemSubmitPerms(邮件只收来自于)，必须用 putEx 才能实现设置和修改，这在后面将提到。关于 put 和 putEx 的详细用法请参考相关文献。

3.3 组(Group)管理

组是组织下面的更小的容器，所以组可以认为是组织的下属单位，比如上面讲到的组织贸易部的下属单位可能会有 MIS、采购等等，那么 MIS 就可以作为一个组。当然，MIS 的下属单位也可以作为一个组，比如 ERP 组。它们都可以是组，只不过 ERP 隶属于 MIS，在后面我们将谈到组和用户的属性"隶属于"。查询组 srchGroup(grpName)和绑定组 bindGroup(grpName)在上文中已谈到。

I. 创建组

将程式 3-3 方法 Create 的第一个参数改成"Group"，如 objFatherOu.Create("Group", "CN= ErpTeam1")，属性设置也作相应得改动，当然，组也不需要设置密码和账户属性，这样就可以创建一个组了。这里要注意的是组类型有两个值分别是："Distribution"、"Security"，组作用域有三各值分别是："Universal"、"Grobal"和"Domain Local"。组的这两个属性的组合为一个 LDAP 名称--"groupType"，它是一个整型值。因为 Win2003 中，如果组

类型为"Security",那么其作用域就不能为"Grobal",所以"groupType" 有五个组合值:Security 和 Universal、Security 和 Domain Local、Distribution 和 Universal、Distribution 和 Domain Local、Distribution 和 Grobal, 分别对应的值是:-2147483644、-2147483646、8、2、4。

II. 组的删除也是通过父组织对象的方法 Delete()来实现的,如:objFatherOu.delete "Group","CN=ErpTeam"。

3.4 组的成员(members)管理

I. 成员添加

组的成员与账户(组或用户)的隶属于是相对应的,因为某个账户添加为某个组的成员之后,则前者也就隶属于后者了。在 ADSI 中是通过组对象的方法 add 添加其成员的,示例代码如程式 3-4:

```
1. Function belongtoAdd(grpName,acctName)
2. Set objGroup = bindGroup(grpName)
3. Set objAcct = bindUser (acctName)' 或者 Set objAcct = bindGroup(acctName)
4. acctPath = objAcct.distinguishedName
5. objGroup.add("LDAP://"&acctPath)
6. objGroup.SetInfo
7. End Function
```

程式 3-4 添加组的成员

II. 隶属于删除

用组对象的方法 Remove 删除组的成员, 比如将程式 3-4 第 5 行改成 objGroup.Remove ("LDAP://"&acctPath)就可以删除该成员了。

3.5 邮箱(MailBox)管理

对账户邮箱的管理主要包括创建和删除邮箱和、邮件只收来自于和邮件转发地址的管理。

3.5.1 用户邮箱管理

I. 创建邮箱

用户邮箱创建是通过用户对象的方法 createMailBox 来实现的。

```
1. Function crtUserMailbox(usrName)
2. Set objMailbox = bindUser(usrName)
3. Set domain = getDm()
4. objMailbox.CreateMailBox ("CN=邮箱存储 (VIRTUALPC),CN=第一个存储组,CN=InformationStore,CN=VIRTUALPC,CN=Servers,CN=第一个管理组,CN=Administrative Groups,CN=第一个组织,CN=Microsoft Exchange,CN=Services,CN=Configuration,"&domain)
5. objMailbox.SetInfo
6.objMailbox.Put "mail", usrName & "@Microsoft.COM"
7. objMailbox.SetInfo
8. End Function
```

程式 3-5 创建用户邮箱

对 AD 较熟的人会发现创建用户邮箱时, 邮件地址并没有创建,但是在删除邮箱的同时,邮件地址却被清空。所以为了保持它们的一致性,在邮箱创建的同时设置邮件地址。

II. 删除邮箱

用户邮箱删除是通过用户对象的方法 DeleteMailBox 来实现的。

```
1. Function deleteUserMailbox(usrName)
2. Set objMailbox = bindUser(usrName)
3. objMailbox.DeleteMailBox
4. objMailbox.SetInfo
5. End Function
```

程式 3-6 删除用户邮箱

用户邮箱删除的同时,AD 会自动将用户的邮箱地址删除。

3.5.2 组邮箱管理

I. 创建邮箱:组邮箱的创建和用户邮箱的创建有所不同,是通过组对象的方法 MailEnable 来实现的, 如 objGroup.MailEnable。

II. 删除邮箱: 在 ADSI 中为了保持与组邮箱的创建方法的一致, 组邮箱的删除是通过组对象方法 MailDisable 来实现的,如 objGroup.MailDisable。

3.5.3 邮件只收来自于(dLMemSubmitPerms)管理

I. 增加邮件只收来自于

dLMemSubmitPerms 是账户的一个多值属性, 其数据类型是数组,所以对它的操作必须用 putEx()方法,如程式 3-7。

```
1. Function addMailfrom(AcctName,Acctarray)
2. Set objUser = bindUser (AcctName) ' 或者 Set objGroup = bindGroup(AcctName)
3. objUser.PutEx 3,"dLMemSubmitPerms",Array(Acctarray)
4.objUser.SetInfo
5. End Function
```

程式 3-7 增加邮件只收来自于

II. 删除邮件只收来自于,如程式 3-8

```
1. Function deleteMailfrom(AcctName,Acctarray)
2. Set objUser=bindUser(AcctName) ' 或者 Set SetobjGroup = bindGroup(AcctName)
3. objUser.PutEx 1,"dLMemSubmitPerms",Array(Acctarray)
4. objUser.SetInfo
5.End Function
```

程式 3-8 增加邮件只收来自于

III. 另外利用 putEx 可以清空和替换该属性的值, 分别是:objUser.PutEx 0,"dLMemSubmitPerms",vbnull) 和 objUser.PutEx 2,"dLMemSubmitPerms",Array(new_Acctarray)。可以看出 putEx 的第一个参数可以有 4 个值 0、1、2 和 3,分别对应清空、删除、替换和增加。

3.5.4 邮件转发地址(altRecipient)管理

在 AD 对象中,只有用户才有邮件转发地址这个属性。对邮件转发地址的操作跟邮件只收来自于不同, 因为前者是单值属性,而后者是多值属性,所以该操作是可以通过 Put()方法来完成。

I. 邮件转发地址添加或修改,如程式 3-9

```
1. Function addMailrec(usrName,AcctName)
2. Set objUser = bindUser(usrName)
3. Set objAcct = bindUser (AcctName) ' 或者 Set objGroup = bindGroup(AcctName)
4. objAcctPath = objAcct.distinguishedName
5. objUser.Put "altRecipient", objAcctPath
6. objUser.SetInfo
7.End Function
```

程式 3-9 增加邮件只收来自于

另外, 邮件转发地址的添加或修改也可以通过 putEx 来实现。

II. 邮件转发地址删除通过 objUser.PutEx 1," altRecipient ", vbnull 来实现。

结束语 这篇文章所谈到的技术只是对 AD 操作中的一小部分而已, 但是这些操作都是作为一个域服务器管理人员经常要做的主要的工作。此外,特别要注意在文中虽然没有提到出错处理,但是它是非常重要的,为了减少篇幅,这里没有提到,请读者自己去添加。

我撰写这篇文章的目的, 就是想给热衷这方面工作的程序员提供一点帮助,虽然不多,但是我会继续研究 AD,并把研究所得和大家一起分享。

参考书目:

1.《The powerful combination of Windows Script Host and Active Directory ServiceInterfaces under Windows 2000》Prepared by Alain Lissoir

2.《Active Directory Service Interfaces SDK》

3.《Windows2000 活动目录开发人员参考库》第 1 卷,(美)David Lesminger 主编,李晓军、邢丽颖、许力等译。机械工业出版社,2001

4.《Windows2000 活动目录开发人员参考库》第 2 卷,(美)David Lesminger 主编,李晓军、邢丽颖、许力等译。机械工业出版社,2001

5.《Windows2000 活动目录开发人员参考库》第 3 卷,(美)David Lesminger 主编,李晓军、邢丽颖、许力等译。机械工业出版社,2001

6.《Windows2000 活动目录开发人员参考库》第 4 卷,(美)David Lesminger 主编,李晓军、邢丽颖、许力等译。机械工业出版社,2001

7.《Windows2000 活动目录开发人员参考库》第 5 卷,(美)David Lesminger 主编,李晓军、邢丽颖、许力等译。机械工业出版社,2001。

8.《Windows 2000 Active Directory 程序设计》(美) Charles Oppermann 著;王磊,王毳等译。

9.《Windows Server 2003 活动目录从入门到精通》(美) Robert R. King 著,薛菲,王曼珠等译,北京:机械工业出版社,2002.1

10.《理解活动目录服务》(美) Daniel Blum 著,王晔译,北京:科学出版社,2003。