HEY, SCRIPTING GUY!

Late-Night Scripting

The Microsoft Scripting Guys

If you've ever watched a so-called B movie on late night TV, you're probably familiar with the following scene. Our heroes are looking for something and, just minutes into the film, they find it. "Man, that was easy!" exclaims one of our heroes, usually the expendable one whose screen time is just about up. "Yeah," says the smarter hero, staring off into the darkness. "A little too easy…."

Needless to say, it's right about this time that our heroes discover they've been set up and the whole thing is a trap. As they try to fight their way to safety to regroup and continue with the rest of the film, the moral of the story becomes clear: if something seems too good to be true, it probably is too good to be true.

Fortunately, script writers don't have to worry about such things. For example, suppose you want to create a new group in Active Directory. Sounds hard, doesn't it? Rubbish! For example, here's a script—four whole lines of code—that creates a group named Managers in the Finance OU in fabrikam.com:

```
Set objOU = GetObject _
    ("LDAP://ou=Finance,dc=fabrikam,dc=com")

Set objGroup = _
    objOU.Create("Group", "cn=Managers")

objGroup.Put "samAccountName", "Managers"
objGroup.SetInfo
```

As you can see, the script starts out by binding to the Finance OU in Active Directory. Once that connection is made it then uses the Create method to create a new group. The Create method requires two parameters: Group (which simply specifies the type of object you want to create) and cn=Managers (which is required because cn happens to be one of the two mandatory attributes that all groups must have).

Of course, when you call the Create method you initially create the group in local memory only; the new account is not automatically written to Active Directory. To do that you need to assign a value to the sAMAccountName attribute; this is the second of the two mandatory attributes that each group must have. Following that, simply call the SetInfo method, which creates the new group and its attributes in Active Directory. If you open up Active Directory Users and Computers you'll see your new group in all its glory.

Man, that was *easy*! A little *too* easy….

If this really was a B movie, now would be the time when the bad guys, a giant ant or a bunch of crazy people armed with Macintoshes would spring out of the bushes, causing everyone in the cinema to jump with fright. Without a budget for special effects, the best we can do here to scare you is pose the following scenario. When you ran your script, you created a global security group. Good for you. But what if we wanted a domain local security group, or a universal security group, or maybe even a global distribution group? What then?

Cue the ominous music. Hold on!

Listen, we apologise for giving you such a fright. Would it help if we told you that all you need to do is add a little bitwise logic to your script and then you can create any kind of group you want? That's OK. We give that advice to lots of people and they all have pretty much the same reaction. We're used to it.

Of course, most of the time the people we give that advice to are asking for help fixing a flat tyre or carrying a load of heavy boxes. In this case, though, our advice actually makes sense. Or at least it will once we finish explaining it to you.

Before we launch into lecture mode, however, let's do a quick review of Active Directory groups. To begin with, there are two broad categories of groups in Active Directory: security groups and distribution groups. The difference between the two? Security groups can be granted access to resources; distribution groups cannot. In other words, you can give a security group, say, read/write access to an object; a distribution group cannot be given access of any kind to an object. Full stop. (In case you're wondering, distribution groups are typically used as e-mail mailing lists.)

The Scripting Guys work for—well, are employed by—Microsoft. When not playing/coaching/watching baseball (and various other activities) they run the TechNet Script Center. Check it out at www.scriptingguys.com.

Figure 1 Creating a Universal Security Group

```
Const ADS_GROUP_TYPE_UNIVERSAL_GROUP = &H8
Const ADS_GROUP_TYPE_SECURITY_ENABLED = &H80000000

Set objOU = GetObject("LDAP://ou=Finance,dc=fabrikam,dc=com")

Set objGroup = objOU.Create("Group", "cn=Managers")
objGroup.Put "samAccountName", "Managers"

objGroup.Put "groupType", ADS_GROUP_TYPE_UNIVERSAL_GROUP _
    OR ADS_GROUP_TYPE_SECURITY_ENABLED

objGroup.SetInfo
```

Figure 2 Possible Group Type Values

Group Type	Constant	Value
Global Group	ADS_GROUP_TYPE_GLOBAL_GROUP	&H2
Domain Local Group	ADS_GROUP_TYPE_DOMAIN_LOCAL_GROUP	&H4
Universal Group	ADS_GROUP_TYPE_UNIVERSAL_GROUP	&H8
Security Group	ADS_GROUP_TYPE_SECURITY_ENABLED	&H8000000

Within each of these broad categories you can create three types of groups:

Global Groups All the users must come from the same domain.

Domain Local Groups Members can be drawn from any domain in the forest, but permissions can be granted to only the local domain (that is, the domain where the group account resides).

Universal Groups Members can be drawn from any domain in the forest and they can be granted permissions anywhere in the forest.

If you have only a single domain in your Active Directory then the different group types don't really matter much (although you still need to be concerned with security groups versus distribution groups). In a multi-domain forest, however, group types become very important. Suppose the Managers group we created a little while ago is for all the managers in our forest, managers from the Europe and Asia domains as well as from the North America domain. That's a problem—after all, our script created a global security group, and in a global group all the members must come from the same domain.

Uh-oh. There's that music again. Trust us—in a B movie, that's never a good sign. But wait, don't panic. Remember, B movies always have happy endings. As it turns out we can use a script to create any type of group we want. All we need to do is set the value of the groupType parameter. And to do that we just need to add a little bitwise logic to our script.

See? We told you that this advice would make sense!

For example, suppose we wanted to create a universal security group. Let's take a look at the code, shown in **Figure 1**, that does this and then we'll explain how it all works.

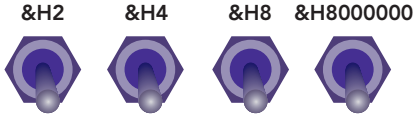
To begin with, we define a pair of constants: ADS_GROUP_TYPE_UNIVERSAL_GROUP and ADS_GROUP_TYPE_SECURITY_ENABLED. The names we give to these constants aren't particularly important; we use the same names found in the Active Directory Service Interfaces (ADSI) SDK at msdn.microsoft.com/library/en-us/adsi/adsi/ads_group_type_enum.asp. The values, however, are very important. To show you what we mean, the different values we can assign to groupType are shown in **Figure 2**.

Got that? If we want to create a universal security group then all we need to do is set the value of the groupType property to &H8. Oh, and we need to set the value of the groupType property to &H8000000.

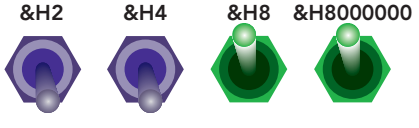
Next we—what's that? Do you have a question? Oh, right: how can we set the value of a property to &H8 and set the value of a property to &H8000000? Looks like we better take a quick side trip into the wonderful world of bitwise attributes.

Bitwise and Pound-Foolish

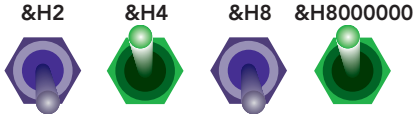
To begin with, we should note that bitwise attributes were not designed to drive you crazy; they were designed to allow a single attribute to hold multiple values. (The fact that they do drive people crazy should be considered a minor side effect.) When dealing with bitwise attributes we find it useful to visualise a bitwise attribute (such as groupType) as being like a control panel with a series of switches. In the case of groupType those switches will equate to the allowed values:



Now, suppose we have a universal security group. To be a universal security group two things must be true: the &H8 switch must be on and the &H8000000 switch must be on. In other words, our control panel needs to look like this:

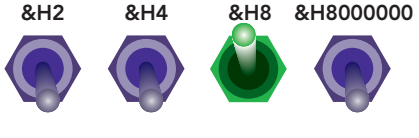


See how that works? What kind of group do we have here?



Exactly. The &H8000000 switch is on, which means we have a security group rather than a distribution group. And because the &H4 switch is on, we know that this is a domain local group. Our final answer: a domain local security group.

You're right: this is more fun than working, isn't it? OK, one more:



Excellent. The &H8000000 switch is off, therefore this is not a security group. Meanwhile, the &H8 switch is on. There's only one possible answer: this is a universal distribution group.

Now that we know how multiple values can be stored in a single bitwise attribute all we have to do is figure out how to actually assign two different values to a single attribute. That's something we'll talk about as we resume explaining the script in full.

And Now for the Exciting Conclusion

After defining the two constants, our script then binds to the OU where we want the new group created. That's what happens here:

```
Set objOU = GetObject _
    ("LDAP://ou=Finance,dc=fabrikam,dc=com")
```

Once the connection is made we call the Create method, passing two parameters: group (the type of object we want to create) and cn=Managers, the cn for the new group. Following that we use the Put method to assign a value to the sAMAccountName attribute:

```
objGroup.Put "samAccountName", "Managers"
```

That should look familiar to you; if it doesn't, simply rewind this column back to paragraph 5 or so. But now comes the tricky part. Here's the code that sets the group type to a universal security group:

```
objGroup.Put "groupType", _
    ADS_GROUP_TYPE_UNIVERSAL_GROUP _
        OR ADS_GROUP_TYPE_SECURITY_ENABLED
```

This is bitwise logic (and, as you might expect, there really isn't anything particularly logical about it). The first part of the code is pretty straightforward: we call the Put method followed by the attribute (groupType) that we want to assign a value to. However, the second parameter—the actual value being assigned to groupType—is a bit unusual, to say the least:

```
ADS_GROUP_TYPE_UNIVERSAL_GROUP _
    OR ADS_GROUP_TYPE_SECURITY_ENABLED
```

As we already determined, in order to make a universal security group two switches must be set: &H8 and &H8000000. At the beginning of the script we created a pair of constants to represent those two values. Now we're using bitwise logic to enable both of those switches for this attribute. To do that we have to specify both constants, joining the two using a logical OR.

Yes, common sense would suggest that you should join these constants together using AND. But, for better or worse, AND has a different meaning in bitwise logic.

Hey, it's a grade-B movie; you can't expect everything to make sense!

See how that works? Suppose we wanted to make this into a universal distribution group. A universal distribution group has only a single switch enabled: &H8. Hence we use code that assigns only a single value (again using the constant ADS_GROUP_TYPE_UNIVERSAL_GROUP) to the groupType attribute:

```
objGroup.Put "groupType", _
    ADS_GROUP_TYPE_UNIVERSAL_GROUP
```

What if we wanted to make a domain local security group? That requires two switches: &H4 and &H8000000. Assuming we've defined a constant named ADS_GROUP_TYPE_DOMAIN_LOCAL_GROUP (with a value of &H4) our code would look like the following:

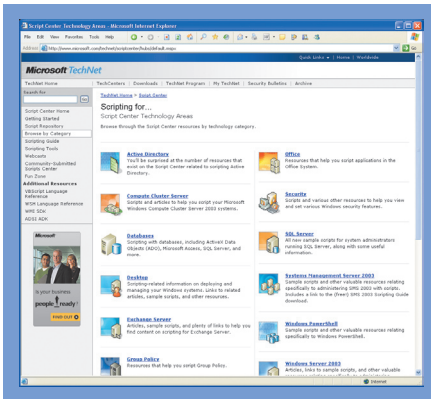
```
objGroup.Put "groupType", _
    ADS_GROUP_TYPE_DOMAIN_LOCAL_GROUP _
        OR ADS_GROUP_TYPE_SECURITY_ENABLED
```

We're not saying it isn't a little weird. But it works, and as long as you squint your eyes a little and tilt your head sideways, it even makes sense. Sort of.

On the bright side, once you understand how the groupType attribute works (and what the acceptable values are) you can do other things, such as determine the group type for an existing group, or even change the type of an existing group (subject to certain Active Directory limitations, that is). For sample scripts that carry out these tasks, take a look at the *Hey, Scripting Guy!* archive in the TechNet Script Center at microsoft.com/technet/scriptcenter/resources/qanda/ad.mspx. And while you're at it, there's a lot more to check out at the Script Center. You'll find not only our old columns, but also tools, puzzles, language references, even sweepstakes! And lots of scripts. For help in finding what you need, see "Your Search is Over" on this page.

As for the Scripting Guys, we're well aware that as stars of a B movie any moment now we'll be whisked off into the sunset with our breathtakingly beautiful new true love. Yep, any moment now. Sorry we can't talk more about group types, but we'll be leaving here soon. Any second now. You know, the traffic can get kind of bad around Seattle this time of night, so it might take a few minutes. But pretty soon we'll be on our way. Yes, pretty soon….

Your Search is Over



Script Center Audio Tour

You're wandering through a maze, every passageway leading into many others. There's no end, never a way out, and you don't seem to ever find what you're looking for. Does this sound like a nightmare? Or does it just sound like your most recent foray into a typical Web site?

Well, we sympathise. And we have something that can help—at least for when you're at the Script Center. The next time you come to the Script Center (microsoft.com/technet/scriptcenter), simply download the new audio tour and transcript. Your friendly tour guide will help you navigate the maze of Web pages and seemingly endless treasure trove of information. Who knows, you might finally find that piece of cheese at the end of the maze.

Browse by Category

Do you need to do a search on Active Directory, but don't know where to look for more information? Try looking under Active Directory on the Script Center. But what if you aren't looking for info on Active Directory? What if instead you want to watch a webcast that will teach you how to script Microsoft Office? Then just look under Office in the Script Center.

Seems pretty obvious, doesn't it? The Script Center lets you browse by category, making it much easier to find what you're looking for. So check out the Browse by Category section of the Script Center (microsoft.com/technet/scriptcenter/hubs) and things just might become a lot more obvious. And more categories are being added all the time.