Session E-SECU

Title

Author

Contents

Are you still logging into your server and your workstation as “Administrator”? Does your SQL Server still have just one user – SA – and no password? Are you interested in setting up a web server on your network, but not sure how to make two servers work on the same network? Are you trying to figure out the difference between domains and groups, and knowing when to log into your domain, and when to login into your workstation? Have you given up on the documentation about domains, groups, Kerberos and a dozen other buzzwords that are tossed around? Mystified about providing access to components according to these roles, groups, users??? In this session, you’ll learn how to build a small company network and create security for a variety of users, machines, operating systems and servers.

The ideas behind users, groups, domains, permissions

Many Fox developers started out as single developers on standalone machines. Occasionally they would develop a multi-user application, but leave it to the customer to handle installing the software and data on the network. Indeed, although I built multi-user applications for Novell networks for a decade – I never actually logged onto a Novell network myself in all that time, much less had one myself. The network admin at the customer's site took care of that. I just gave them an EXE and some data files, and we discussed whether exes belonged on the server or on workstations to improve performance. And pointed the exes to a centralized data source so that all the users pointed to the same database.

The mental block that many people have is that when they install Windows NT on a standalone machine, or on one that's connected to a network without a domain (a peer to peer network), the logon information is stored on that machine. Then users on specific machines decide what they're going to share with other users on that network.

The fundamental concept behind Windows NT security is that information about users and permissions is all contained in a single place, and users access that single place. This single place is called a domain. You can still log on to your own machine without access to the domain (hence the cry, "Network is down!"), but that does not get you access to the rest of the network.

Specifically, you set up a computer running NT Server (or Windows 2000 Server) to be a domain controller. That machine is Lord God King when it comes to knowing who can do what. The users and permissions are stored on that machine. What happens if that machine goes down? It would be a good idea to have a backup, right? A second machine to step in and take its place. That's a backup domain controller – and if you have a backup, then the first one is actually the "primary domain controller."

As an aside, typically you don't set up a machine just to be a domain controller and not have it be used for anything else. Most commonly, your server will also be the place where you keep all your common files – in other words, your file server. If you've got a large network, though, you may have data scattered around multiple servers – but only one of them is the primary domain controller. For the sake of this discussion, we'll assume that your file server is also your primary domain controller.

OK, so the domain is where a list of users is stored. But there's more to it than that.

Each user can be given a variety of permissions, or rights, to access directories, files and device on the network. For example, one user may be allowed full access (read/write/delete/modify) to a specific directory while a second user may only have read access, and third user may not have access at all.

Once your network grows past two machines and a couple of users, it will become unwieldy to duplicate these permissions for subsequent users. Typically, you'll be able to group users into similar categories. For example at a small software development company, you will have a couple of partners, several senior software developers, a couple of junior software developers, a test or QA person, and a couple of clerical personnel. The permissions that each of these people need would be similar across job descriptions – the software developers have access to certain directories and devices while the clerical personnel would need access to other directories and devices. And these may overlap, or they may be mutually exclusive.

This is where the concept of groups comes in. Instead of assigning permissions to a user, you create an entity called a group, and then that group is assigned permissions and given access to selected devices. Then you add users to those groups. When new people join the company (really, the network), they are added as a user, and then that user is added to the appropriate group. Their user then automatically has the permissions and access that the group has. It's a lot easier and less time-consuming to set up and maintain.

You still have granular control – if you have to lock a user out, you can do it to that user, but often you'll add or remove a permission for the entire group instead of having to do it to every individual user.

As with the users, the permissions and the groups are maintained on the domain controller.

Setting up a file server as a domain controller

We won't walk you through the actual installation of NT or W2K, but will mention some items that you'll want to keep in mind.

First of all, don't set up a server when you're under a time crunch. Murphy's Law dictates that something will go wrong and it will take three times as long as expected. The install process can take an hour or two, and then you can spend another couple of hours doing tweaking. If you're under the gun, best not to attempt it.

I would also recommend expecting that you'll do your first server install several times. I've installed NT Workstation on the same three boxes so many times that at one point, I could have literally done it without having the monitor on – I knew each dialog and keystroke by heart.

NT 4.0

The first true decisi
on you'll have to make is what to name your server. By this I mean the name of the physical computer that you're installing Windows on. This name is one that other people will be using for a variety of reasons, so make it easy to spell, don't use a risqué or offensive name. If it is possible that you'll eventually have more than one server, you may want to have a commonsense name so people can remember what the server is for. Our first web server, for example, was called WEBBIE.

The next decision you need to make is whether to make it a Primary Domain Controller (PDC), a Backup Domain Controller (BDC), or a regular server that is a member of a Workgroup. If you pick the wrong one, the only way to fix it is to reinstall the operating system. Blech.

You'll be asked to select a password for the administrator of the system. You should pick a very, very good password. One that's hard to break and easy to remember. Typically, a common word with numbers and odd letters scattered in the middle is a good choice. If your first car was a Dodge and you bought it for thirty-two dollars, perhaps dod32ge4me. Get it?

If you forget the administrator password, you get to go back to go, you don't collect $200, and you get to do the install all over again. Remember the magic word: Blech.

The next decision you'll need to make is the file system. There are three common file systems available for NT-style systems: FAT16, FAT32 and NTFS. NT 4 supports FAT16 and NTFS while W2K supports FAT16, FAT32 and NTFS.

FAT16 is the file system that's been around since the pre-Windows era – it handles all of the details of files, directories, and so on. FAT32 showed up near the end of the Windows 95 releases, and provided some slight enhancements and performance improvements. NTFS – New Technology File System – debuted with Windows NT and provided for a wide variety of security functionality. It allowed the operating system to permit and track who has access to what entities on the network.

When you set up a machine as a file server, you'll typically create a couple of partitions – one for the operating system, and another for data files. Some system administrators set up the operating system partition as FAT and the data files as NTFS, arguing that doing so allows them to boot the machine into DOS. Others argue that using NTFS even for the operating system partition provides enhanced security, and that a proper approach to backup will alleviate any need to boot the machine into DOS. As with many seemingly black and white technical issues, this decision can escalate into a religious war; I won't recommend but will let you decide.

You'll be asked about whether you're connected to the network, and what kind of network card you've got.

The next step is to choose the protocol the system will use. While you have several choices, TCP/IP is the one to go with.

You'll be given the choice to install a variety of different services and tools. For the purposes of this discussion, we'll ignore those. If you're hedging on whether to select something, don't do it unless you have a specific reason to do so. You can always add a needed service later.

You'll need to enter the TCP/IP properties – including the IP Address, subnet mask, and default gateway.

After the TCP/IP stuff, you'll be asked for the domain name. Note that (1) this is not the same as a "domain name" on the Internet, and (2) this is different than the machine name for the server. You can kind of think of the domain as the name for the entire network, while the server and each of the workstations have their own names. Again, people will be using the domain name regularly so pick something easy to spell and nothing offensive or obnoxious. If you've got a small company, you could just use a funky name, if you're in a larger corporate environment, you may need to adhere to a larger set of rules, or help people out with a domain like "ACCOUNTING" or "SOFTDEV".

A couple more screens and you're done. Pull the floppies and/or CDs out, reboot, and it'll ask you to log on as administrator.

Windows 2000

Windows 2000 introduced a new mechanism called "Active Directory" which is basically NT4 domains on steroids. It is Microsoft's attempt to address the networking and directory service requirements of larger corporations – as you can imagine, the domain handling and file system mechanisms suitable for a workgroup of a couple hundred users, tens of thousands of files and a half dozen servers is considerably less demanding than for a company of 12,000 users and several million data files spread out over hundreds or thousands of servers.

Many of the steps for install Windows 2000 are similar or the same. Once you finish the initial install, you'll pull out your CD or floppies. Then it will be time to create your Active Directory domain.

You'll need to determine which type of domain installation to use. Basically, you can choose native mode or mixed mode. Mixed mode is required if you have older machines – NT Servers - set up as domain controllers and you want to share security info between your new W2K domain controller and your older NT domain controllers. If this W2K server is your first server, then you will want to run AD in its native mode.

The second major difference is that you need to use a domain name server as part of running an Active Directory domain. (A DNS is a mechanism to translate friendly domain names like www.hentzenwerke.com to the numeric IP addresses like 43.22.109.44 that the Internet actually uses. If you don't have a DNS already, the install process will install a DNS on your server for you.

Along these same lines, when you give your W2K server a domain name, it will expect your Internet domain name. The domain name is the "Hentzenwerke.com" part of www.hentzenwerke.com. What? You don't have one? You can make one up on the spot – it's probably a good idea not to use an existing domain name, though, if you think you'll eventually connect this network to the Internet. If you name your server "Microsoft.com", when you later connect to the Internet, all your machines will get badly confused between your own server and the real "Microsoft.com" elsewhere on the Internet. Badly confused.

After you answer a couple more questions, Windows will attempt to connect you to a DNS to verify the domain you entered. Unless your own DNS is an active functioning DNS (which it isn't at this point) or if you've got access to the Internet, this attempt will fail and give you an error. Take heart, bunky! It's OK. Let Windows install and configure your DNS.

You'll need to determine default permissions for this domain – if this is your first server, select "Permissions compatible only with Windows 2000 servers". Enter the administrator's password, and run out through the next couple of dialogs and finish the setup. Get a cup of coffee – it can take a while.

Setting up a workstation and adding it to the domain

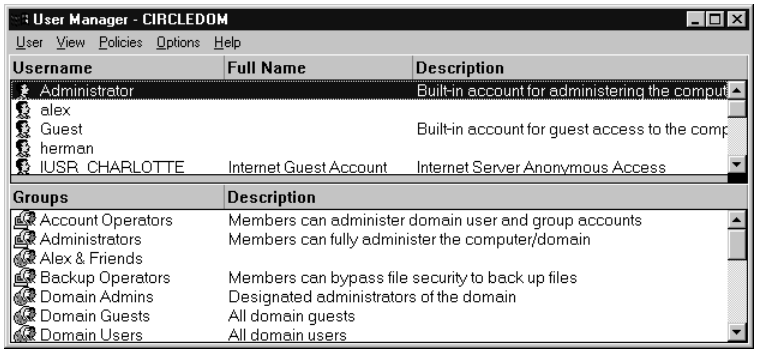
Now that you have a server in working order (you do, don't you? Even if it took more than one try?), it's time to set up workstations that will connect to the server.

It's a lot less trouble to set up a new workstation and connect it to the domain, but most often, you'll already have computers that are in use. For purposes of this discussion, we'll assume that your workstations are all NT or W2K.

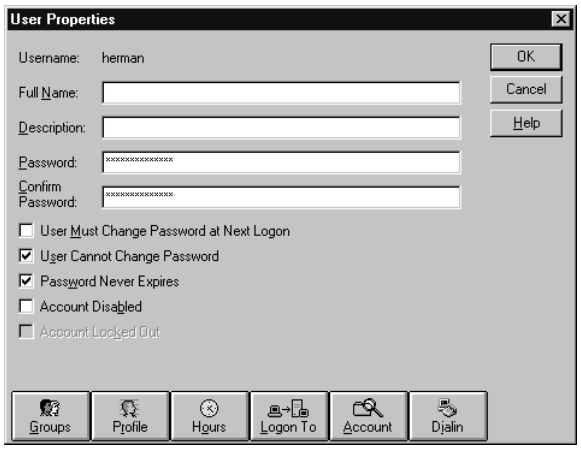
Since you've got an NT workstation, it's already set up with at least one user – administrator. You may also have a second user, such as Bob, set up on that workstation as well. And both the administrator's credentials and Bob's credentials – passwords and rights – are all stored on the workstation's hard disk itself. The first step is to set up a user within the domain, such as WorkerBeeOne, and then, when logging the workstation, selecting WorkerBeeOne on the domain instead of the workstation's administrator or Bob.

Here's how to add a user to the domain on an NT Server:

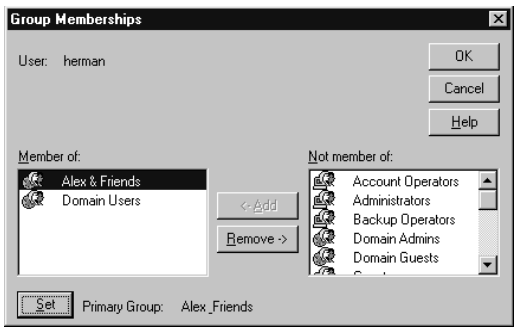
1. On the server, log in as Administrator
2. From the Start Menu, select Program Files | Administrative Tools | User Manager for Domains



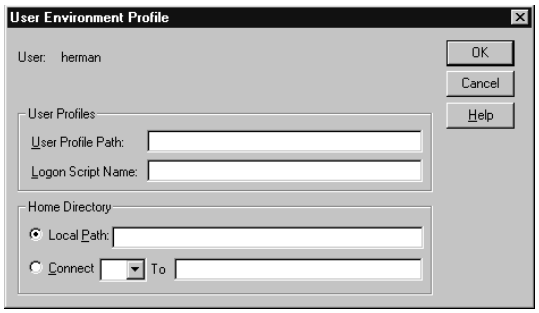
3. Select User | New and open the User Properties dialog.



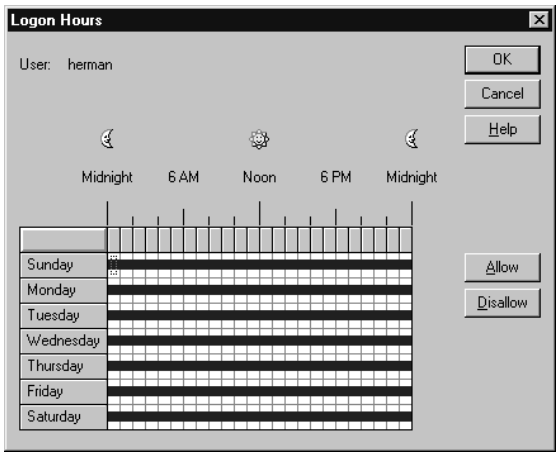
4. The Groups button is used to add or remove the user from a group.



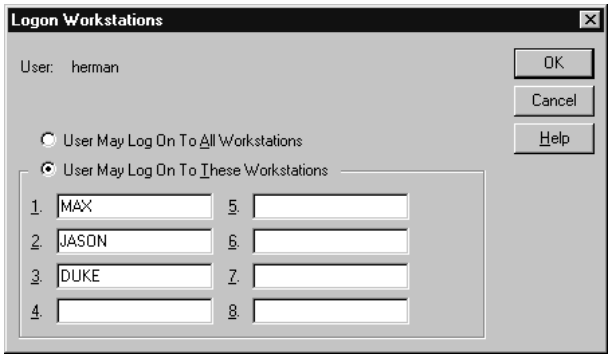
5. The Profile button is used to define profiles for the user, including logon scripts.



6. The Hours button is used to define which hours the user is allowed to log on to the domain.



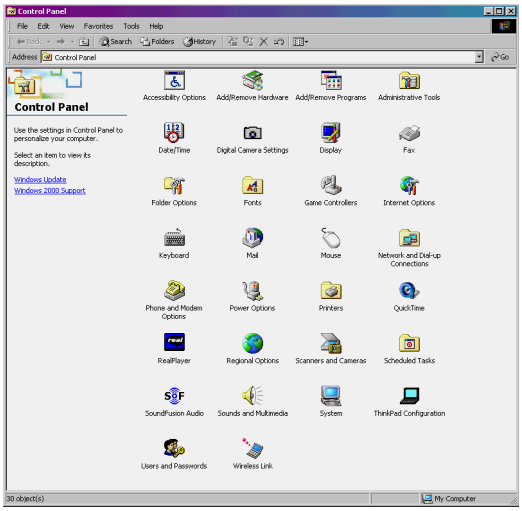
7. The Workstations button is used to define which workstations the user is allowed to log on from.



Here's how to add a user to the domain on a W2K Server:

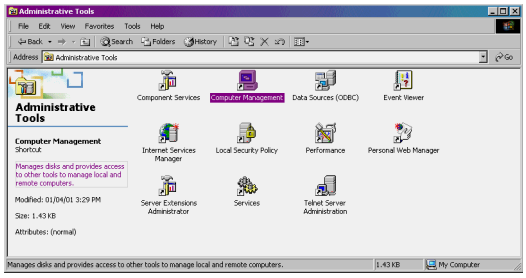
1. Start | Settings | Control Panel

2. Get the Control Panel screen



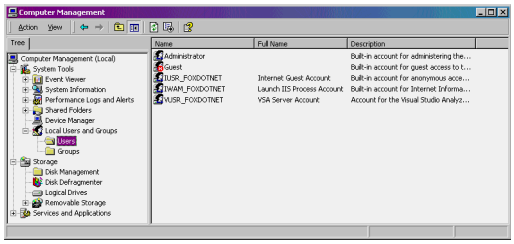
3. Select Administrative Tools applet

4. Select Computer Management

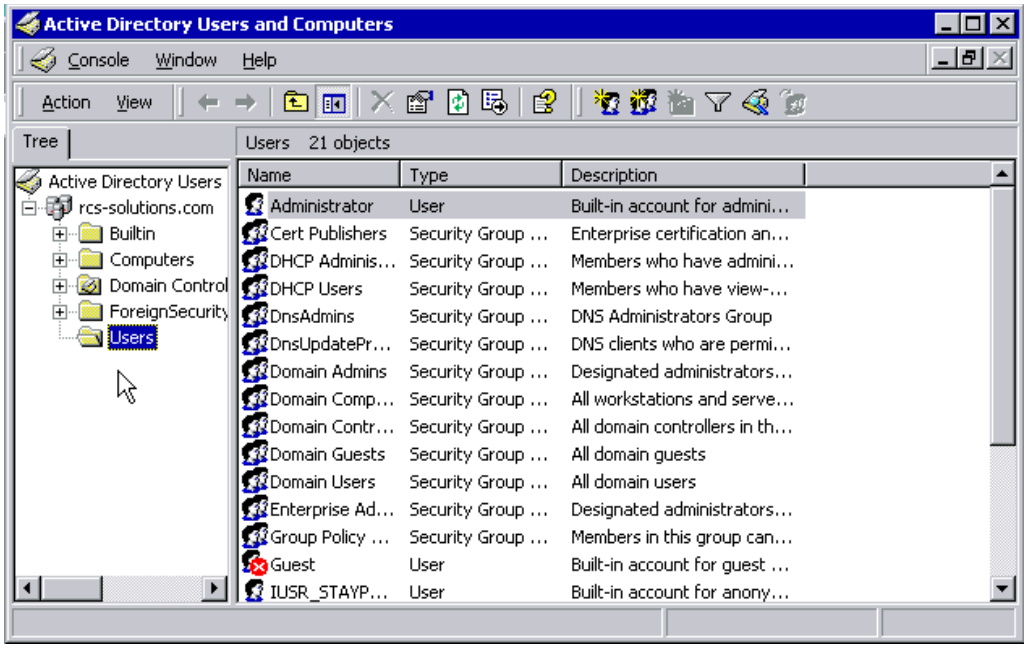


The Computer Management dialog will display

For a non-domain controller, the screen will look like this:



For a domain controller, the computer management dialog will look like this:



In each case, you can highlight the appropriate node and see the users and groups. Right clicking on an object will open a context menu with a Properties menu option that will allow you to change the properties for that object, in a similar fashion to NT.

The next step is to add the workstation to the domain, so that the domain knows about the machine. When the workstation is connected to the network, the machine will detect the existence of the domain, and thus the domain will appear in the combo box. When you enter a domain user and password, the domain will attempt to authenticate the user, but first will not find the machine. A dialog will appear asking you to add the machine to the domain, and request the name and password of an authorized user. What the dialog is asking for is the administrator or other authorized user already on the domain – not on the local workstation.

The last step is to log on to the domain from the workstation. Well, we also have to set up permissions for the user, but we'll do that after we cover groups.

Here's how to log on to a workstation as that user:

1. Ctrl-Alt-Del to get the logon screen.

2. You'll get a screen with two text boxes and a combo box.

3. When you pop open the combo box, it will have at least two entries in it. The first entry is the name of the local workstation. The second entry is the name of the domain.

4a. At this point, you have two choices. If you select the name of the local computer, only user names and passwords for that computer – such as administrator and Bob – will work. And you won't be logged onto the domain, and you won't have access to all of the domain's resources, such as the file server files, other devices, and other machines whose resources are shared.

4b. If you select the name of the domain controller, only user names and passwords that are entered into the domain controller – such as WorkerBeeOne – will work.

Setting up groups and users

Basic concepts – just as you've set up users on your domain, you'll set up groups on the domain as well. This includes creating the group and then assigning permissions to the group. We'll do a couple of simple ones here, so you don't get distracted from the task at hand – groups and users – in the next section, we'll get into permissions in more detail.

The last step having to do with groups and users is to add users to various groups.

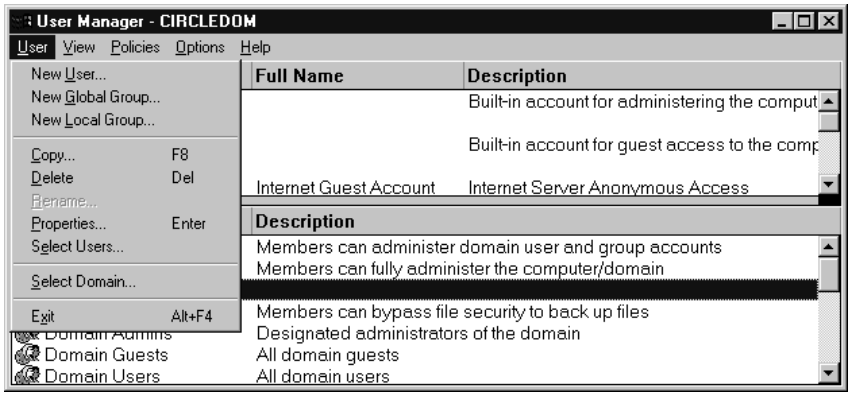
Let's create a simple scenario – suppose we've got a company with three types of employees – the two partners, three software developers of equal rank, and two clerical people of equal rank. So we'll need three groups – JustPartners, Developers, and Staff.

JustPartners will have access to the entire network – after all, it's their company. Developers will have access to a subset of the network – they won't have access to other people's personal files, nor will they have access to company administrative and financial data. And the Staff will have access to SOME of the admin and financial data that the partners have access to, but not all. And they won't have access to the development projects that the developers have access to. And both Developers and Staff would have access to some files together, such as the customer database and the company calendar.

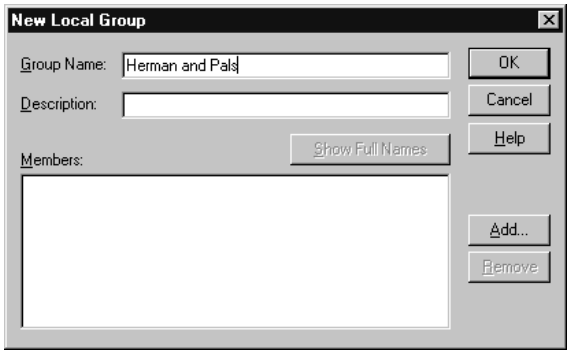
If you drew a Venn diagram (remember those?), Partners would be the big circle encompassing everything, Developers would be one circle inside the big circle, and Staff would be another circle inside the big circle. And the developers and staff circles would slightly overlap. The important thing is that developers and staff both have access to things that the other group doesn't.

Step 1 is to create the groups.

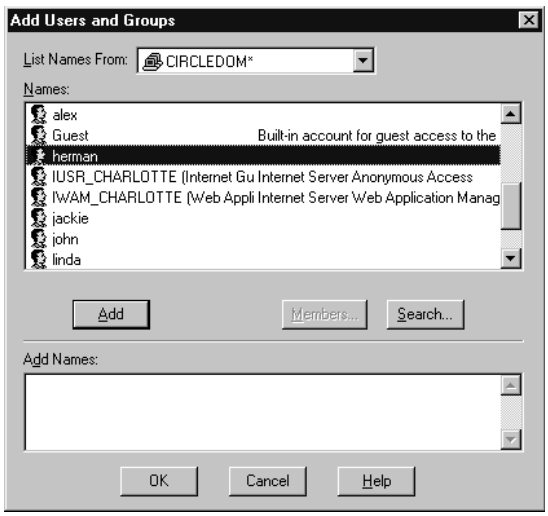
In the User Manager (shown for NT), select the User | New Local Group or New Global Group menu.



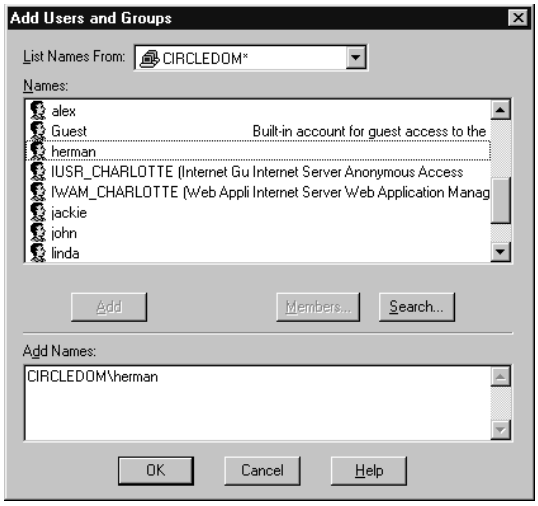
Enter the name of the group. Then Select the Add button to bring forward the Add Users and Groups dialog.



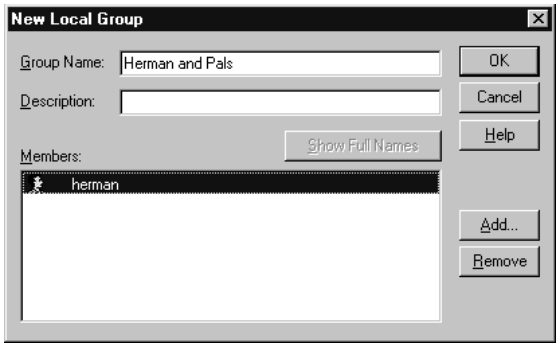
You'll see a list of all available users. Select a user and click the Add button between the two boxes.



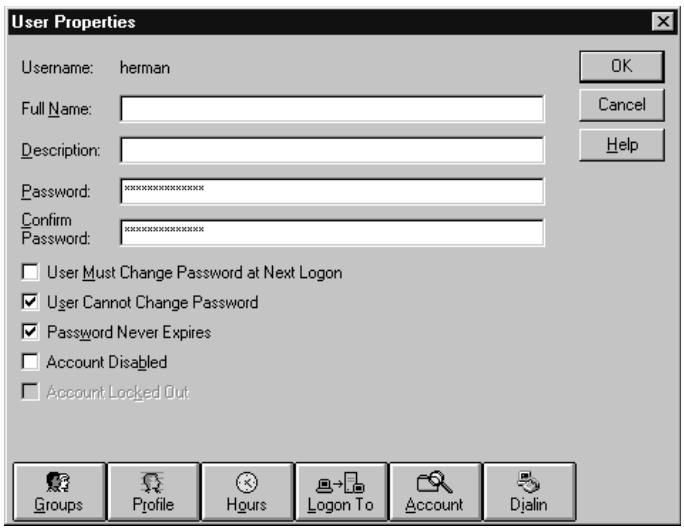
The name will appear in the Add Names box. If you are done, click OK. If you want to add more users to the group, select another user from the Names list box and click Add again. Repeat until light and fluffy.



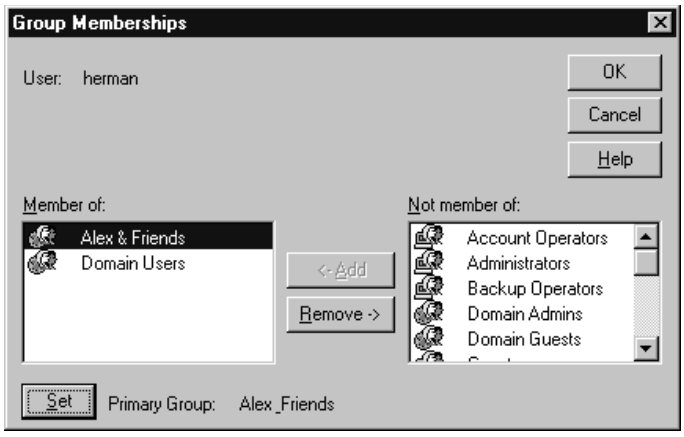
After closing the Add Users and Groups dialog, the New Local Group (or New Global Group) dialog will show the names of the members of the group.



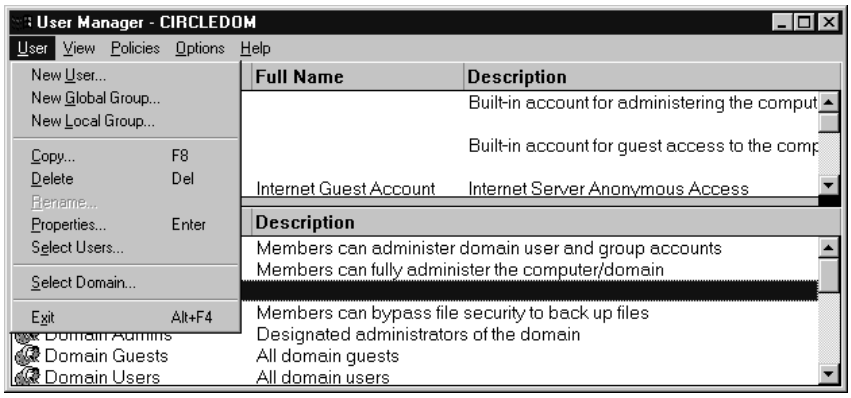
Since a user can be a member of more than one group, you can determine which groups a user belongs to by opening the User Manager for Domains dialog and click on the Groups button in the lower right corner.



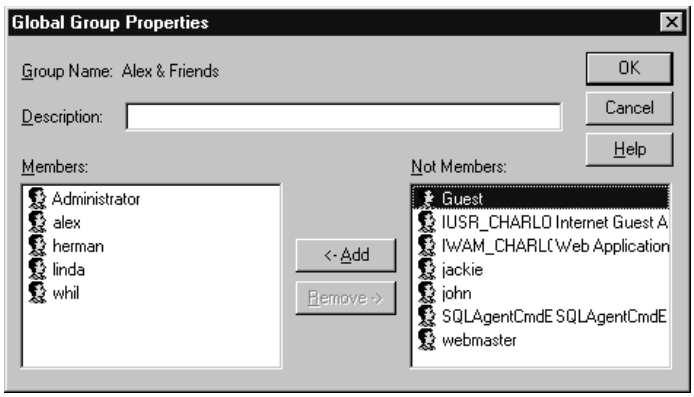
The Groups that the user belongs to will be displayed. You can add the user to other groups easily with this dialog.



Similarly, you can determine the members of a group by clicking on the name of a group (shown under the User menu in this screen shot), and then selecting the User | Properties menu option.



This brings forth the Group properties dialog.

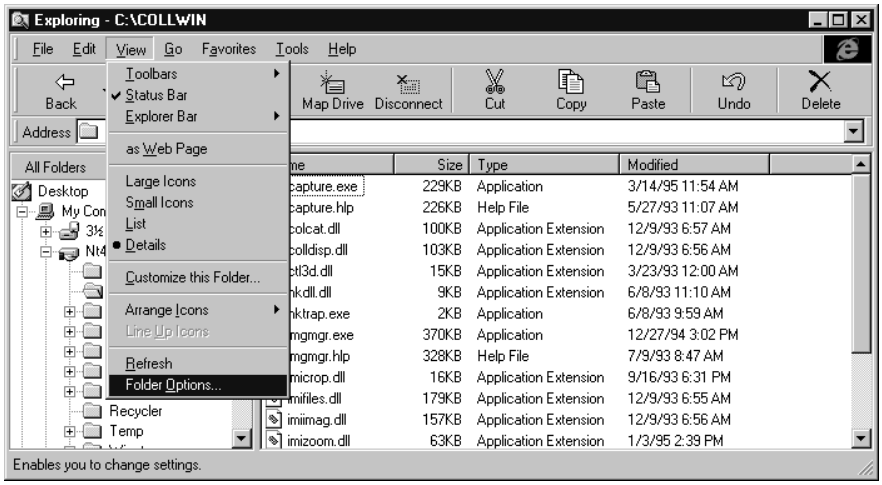


For W2K, select the New Group icon in the tree view in the Computer Management applet, and follow similar steps.

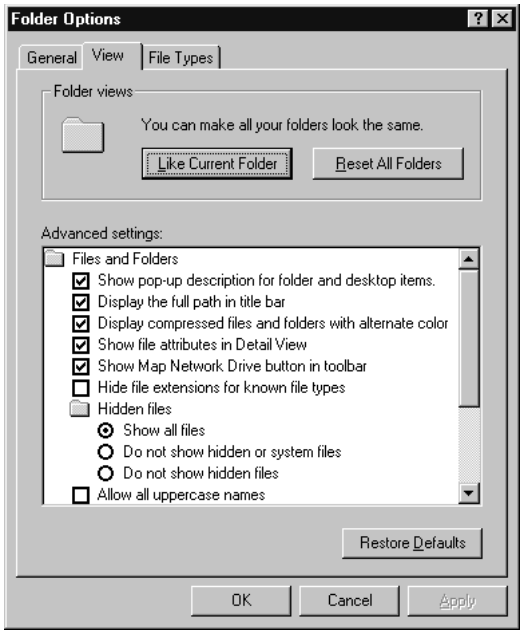
Setting permissions on file server directories

The last step is to set up permissions on directories, files and devices. In this step, you allow users or groups access to perform specific functions on objects of your choosing.

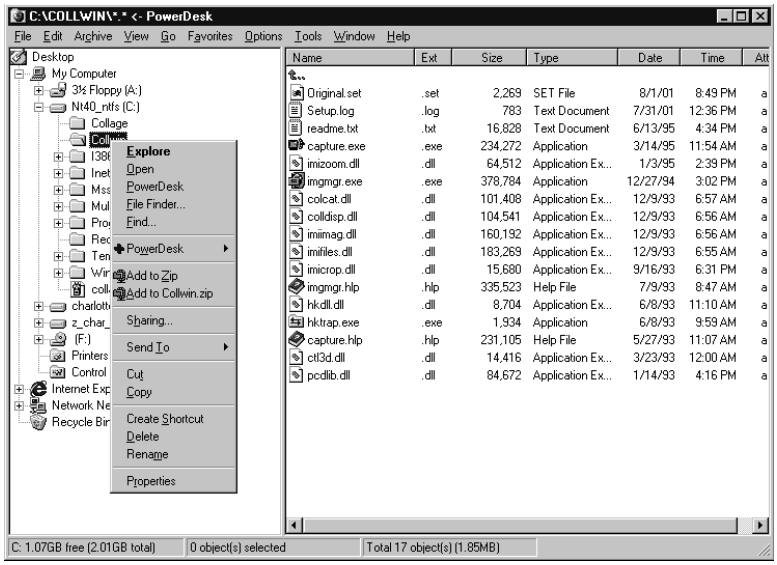
Before you jump into this step, though, let's do a bit of housekeeping first. Open Windows Explorer and select the View | Folder Options menu option to bring forward the Folder Options dialog.



In the Folder Options dialog, select the View tab, and set the settings as shown in the figure. These settings will help you get the best view of what's going on in Explorer.

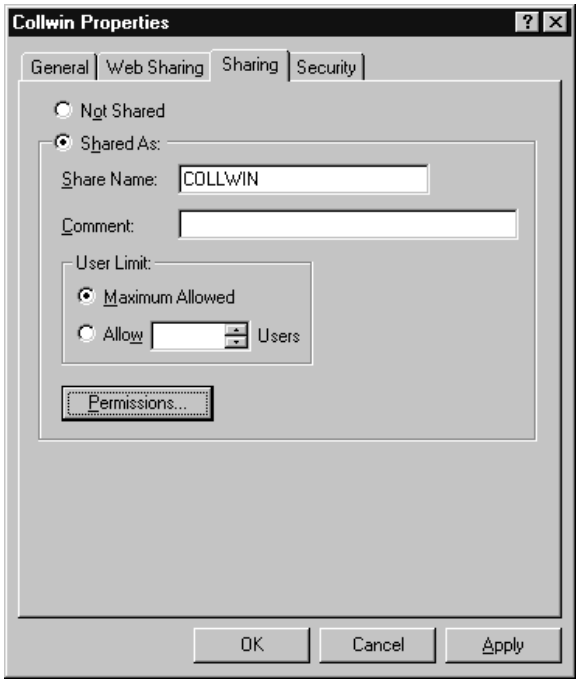


Now, to configure permissions on a specific directory, right click on that directory to display the context menu.

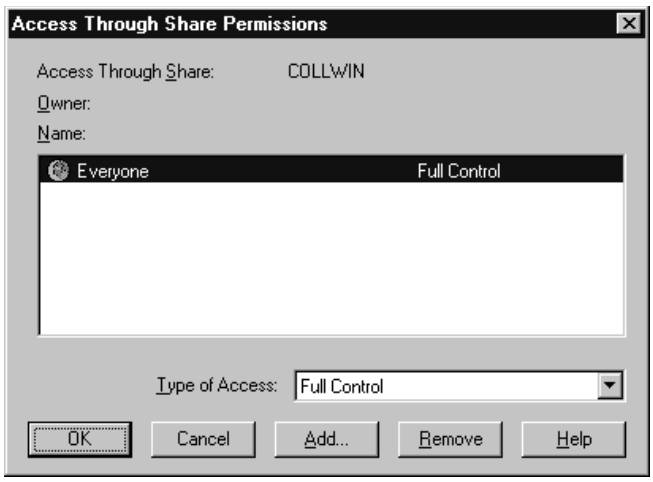


What we're going to do is allow access to the COLLWIN directory on the file server to two different groups of users. The first groups of users will only have Read access while the second group will have Full Control.

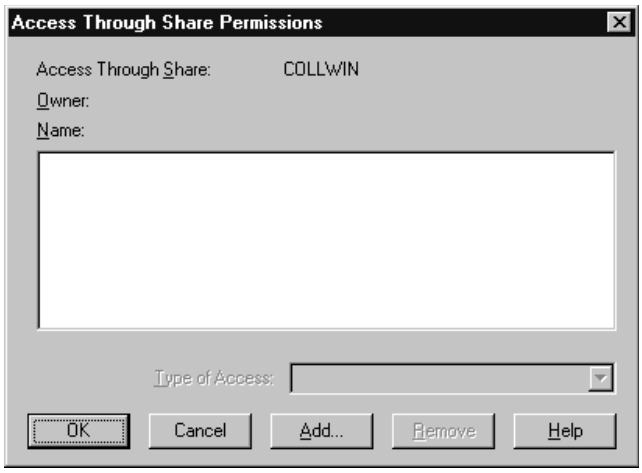
Select the Sharing menu option (or select the Sharing tab in the Properties dialog if you select the Properties menu in the context menu.) Select the Shared As option button, and notice that all the controls become enabled.



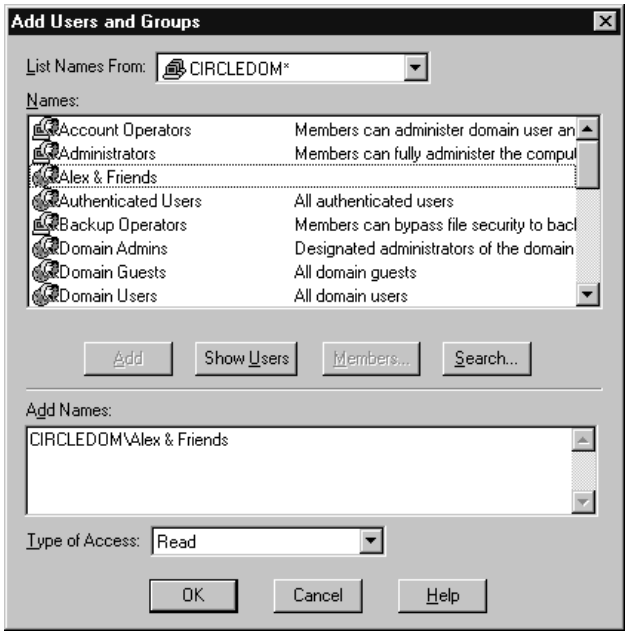
Click the Permission button to bring forward the Access through Share Permissions dialog.



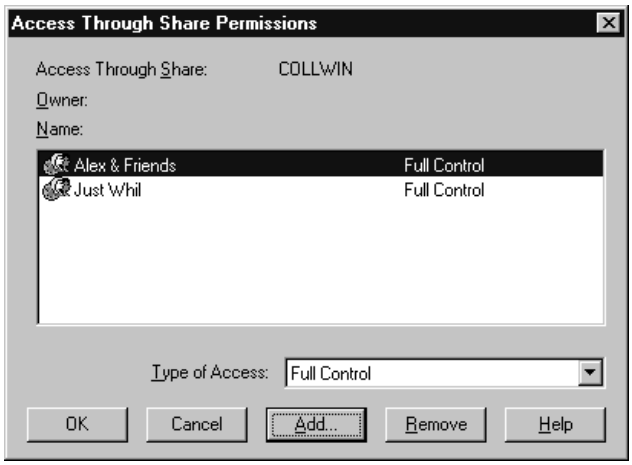
You'll see that, at this point, Everyone has access to the directory. Generally, this is not what you want. The safest security policy is to take away all access, and then specifically grant it to groups who need it. Click the Remove button and now no one will have access to the share.



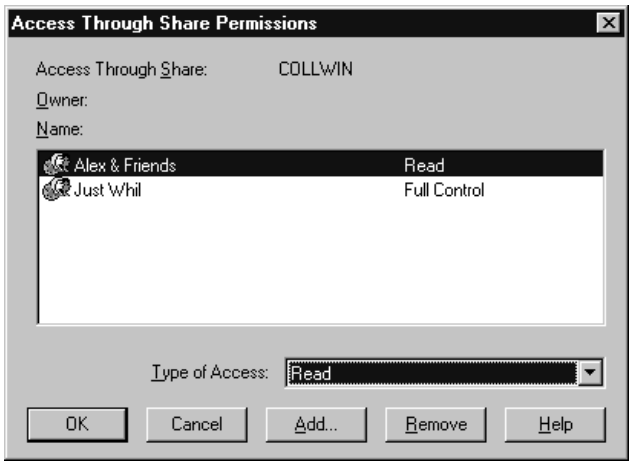
Now it's time to add back in those groups who need access. Click the Add button to bring forward the Add Users and Groups dialog. Select the Names of the Groups you want to have access.



Repeat this step until you have added all the groups. Note that if you change the value in the Type of Access combo box, everyone that you've added to the Add Names edit box will get that same Type of Access.



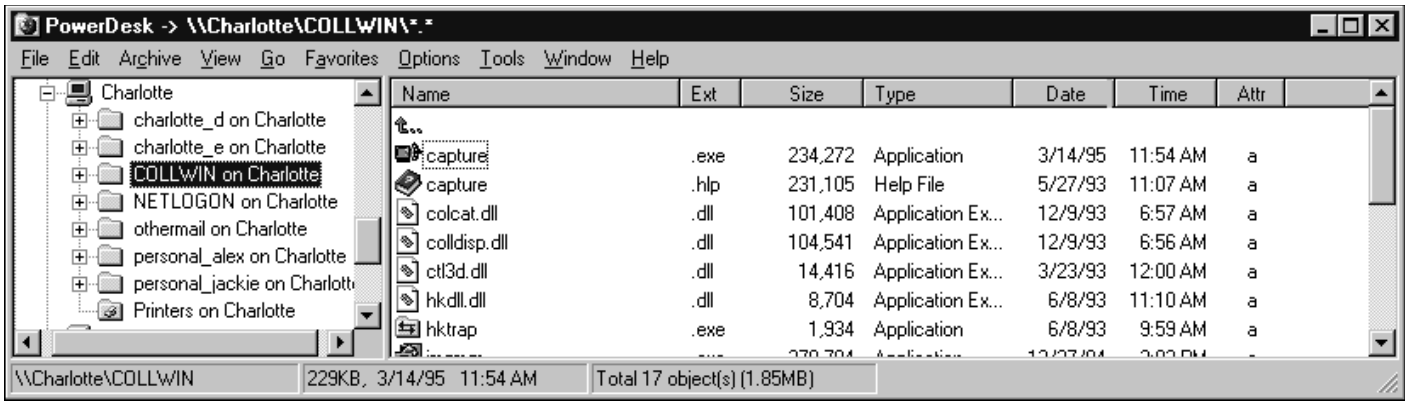
If you want different groups to have different types of access, either add them with the Add Users and Groups in separate steps, or add them all, and then change the Type of Access one by one in the Access Through Share Permissions dialog.



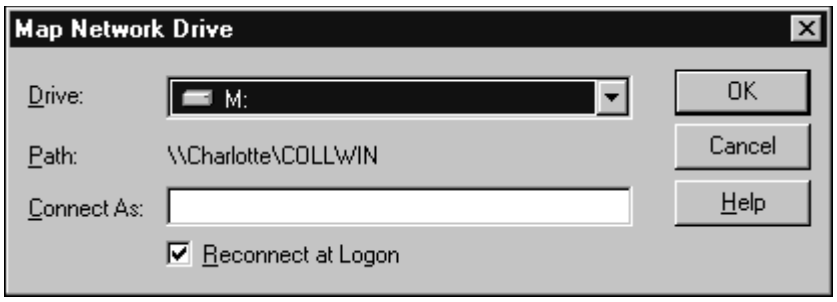
The COLLWIN directory is now accessible through My Network Places in Windows Explorer on workstations on the network. From one of those workstations, drill down into the server and you'll se the name of the share as you specified it earlier in the COLLWIN Properties dialog.

The last step is to map this directory to a drive letter for easier access.

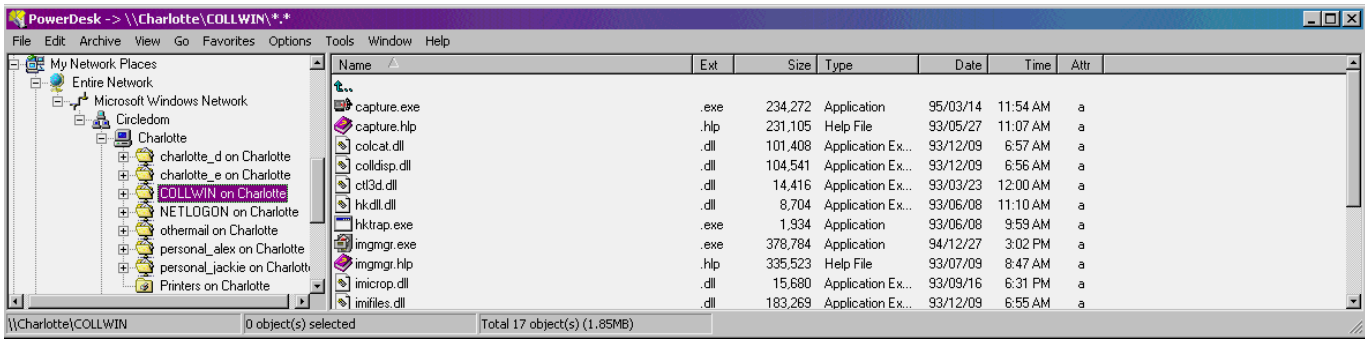
In Windows NT, drill down into Network Neighborhood, right click on the COLLWIN share, and select Map Network Drive.



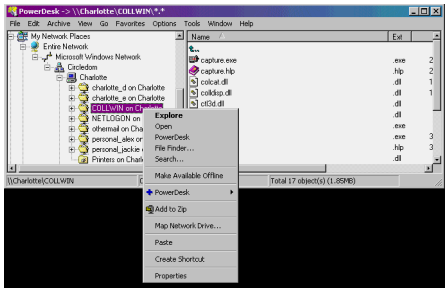
The Map Network Drive displays a dialog that allows you to select a drive letter.



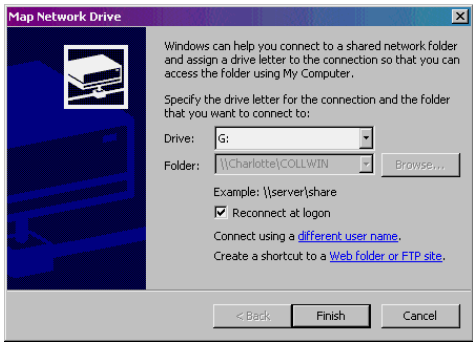
In Windows 2000, open My Network Places and drill down to the COLLWIN share on the server.



Right click on the share and select the Map Network Drive menu option in the context menu.



The available drives will display.



SQL Server

The database application, Microsoft SQL Server, is a collection of Windows Services and a set of applications that are used to manage the server, assist with development and management, and provide access to the data store itself. The data store is composed of several physical files that contain one or more logical entities – databases. And a database is a collection of objects, such as tables. Some tables hold data that end users are interested in, such as customers and auto parts. Other tables are used by the system – such as those that contain users, logins, and so on.

You can't just access a SQL Server database like you can a FoxPro dbf. You can use one of the applications that is included with SQL Server, such as the Query Analyzer, Or you can create a connection to the database, using one of several different mechanisms, such as ODBC or an OLE DB provider.

Whichever way you go, the access request goes through a central SQL Server security module, which enforces (or restricts) access via the data and requirements in the system tables.

There are several levels of security that have to be passed before "getting" to data or doing things.

The first step is that the user must log in to the server and be authenticated. Once authenticated, the user has access to the server.

The next step is for the login to be given access to the database, and mapped to a user within the database.

Finally, the permissions of the user determine what data they are allowed to access or what functions they are allowed to perform.

How these steps are performed depends on whether SS is set up to authenticate via "NT Authentication" or through "SQL Server authentication." When you install SQL Server, you can choose one or the other. (Actually, the choice is either NT AND SQL Server or just NT)

SQL Server Security

There are three pieces to security. First, there are 'logins'. These are combinations of login names and passwords contained in the SQL Server database – specifically, the sysxlogins table in the master database.

Second, there are database users. These are also contained in the SQL Server database – in the sysuser table.

Finally, the database user has a series of permissions for which they are granted or denied. There are two types of permissions - object permissions that give you the ability to perform SELECT, INSERT, UPDATE and DELETE commands, and statement permissions that give you the ability to create and manipulate objects. For example, one database user may only have permission to run SELECTS against a database while another user may have INSERT and UPDATE ability as well. A third user may have permission to not only access data, but to perform operations like creating new databases and perform maintenance routines on existing databases.

Types of Authentication

With SQL Server Authentication, the SQL Server needs to be presented with a login ID and password. Typically, an application that is accessing the SQL Server will display a dialog that prompts the user for the login information. That information is then passed to SQL Server, and the login ID and password are authenticated in the server. If it succeeded, the client then has access to any database that is has been mapped to.

When the user attempts an operation, such as a SELECT, the permission for that database user is looked up in order to determine if the user is allowed to do that.

With NT Authentication, an attempted access to the SQL Server doesn't generate a login dialog. Instead, the user currently logged into the NT workstation or client computer has already been authenticated by Windows. SQL Server trusts the domain, so you can think of the authentication needed by SQL Server as having been doing transparently. If the user is found in the SQL Server SYSXLOGINS table, the user is ordinarily mapped against a database user (but doesn't have to be – for example, an entry can be made in the SYSXLOGINS table denying access a user.) Then, similar to SQL Server Authentication, when the user attempts an operation, such as a SELECT, the permission for that database user is looked up in order to determine if the user is allowed to do that.

Note that authentication – in either case - merely ensures the identity of the client (that logged in.). Authentication does not determine or apply permissions.

Scenarios

1. NT Authentication:

User logs onto machine as user HERMAN
SQL Server is set up as NT Authentication
SQL Server has a HERMAN login that's mapped to the Windows user HERMAN
So HERMAN, once logged onto the machine, can access SQL Server

2. NT Authentication
User logs onto machine as CARL
SQL Server is set up as NT Authentication
SQL Server does NOT have a CARL login that's mapped to the Windows user CARL
So CARL can't get at SQL Server

3. SQL Server Authentication
SQL Server is set up as SQL Server Authentication
SQL Server has a login of DONNA, and password of DONNAPW
User logs onto machine as HERMAN
User accesses SQL Server
SQL Server puts dialog up, asking for login id and password
User enters DONNA/DONNAPW and gains access

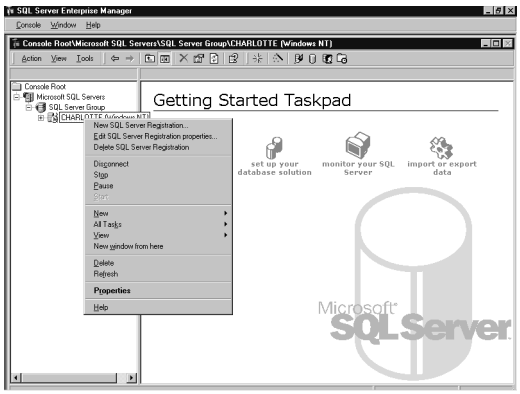
4. Visual FoxPro application
Application is running on a workstation
SQL Server is set up with NT Authentication
MIKE gets on a box as windows user MIKE
Then he walks away (Bad Mike!)
LAURIE logs in into the Visual FoxPro application on that machine, and get to SQL Server via a remote view in the application, say.
Although LAURIE logged into the app as LAURIE, the SQL Server thinks MIKE is the user, because that's who the Windows user is

How to define what type of Authentication:

1. Run Enterprise Manager

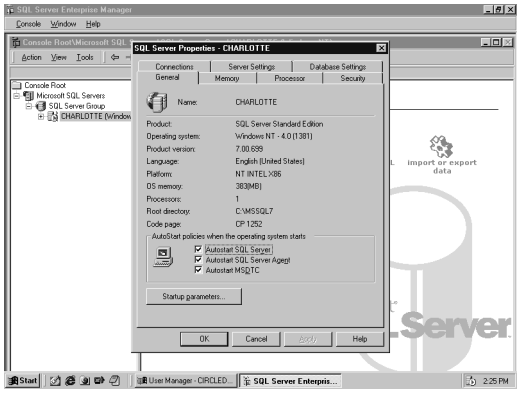
2. Select the SQL Server

3. Right click on the SQL Server

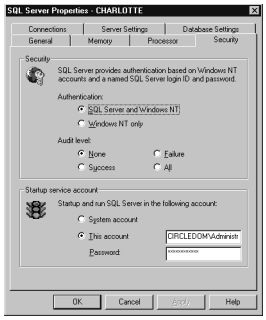


4. Select Properties

5. You'll get the SQL Server Properties dialog



6. Select the Security tab



7. Select the Authentication option button you want.

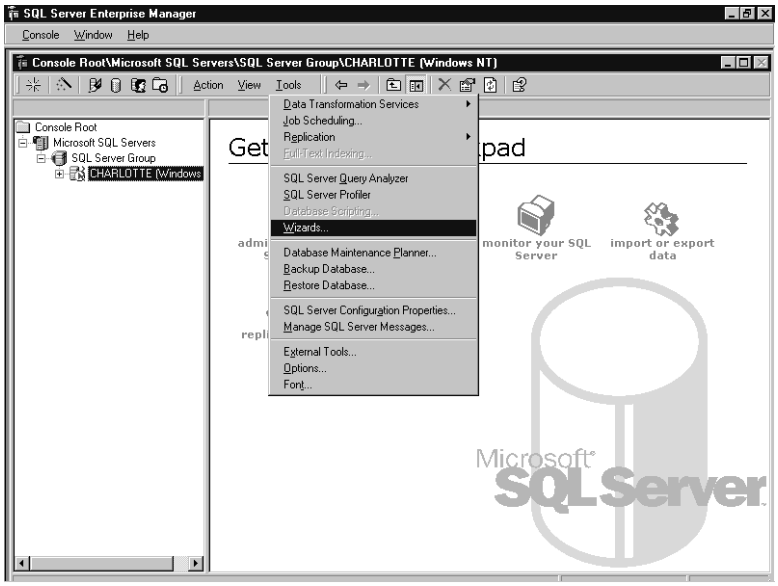
Creating a Windows NT login

1. Enterprise Manager

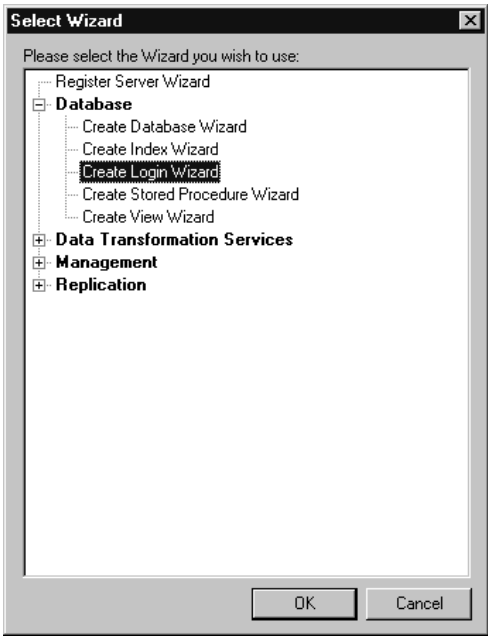
2. Select the SQL Server you want

3. Click on Tools button (to the right of Actions and Views)

4. Select the Wizards menu option. Note that this menu option is not enabled if you don't have a specific SQL Server highlighted in the Console Root tree.

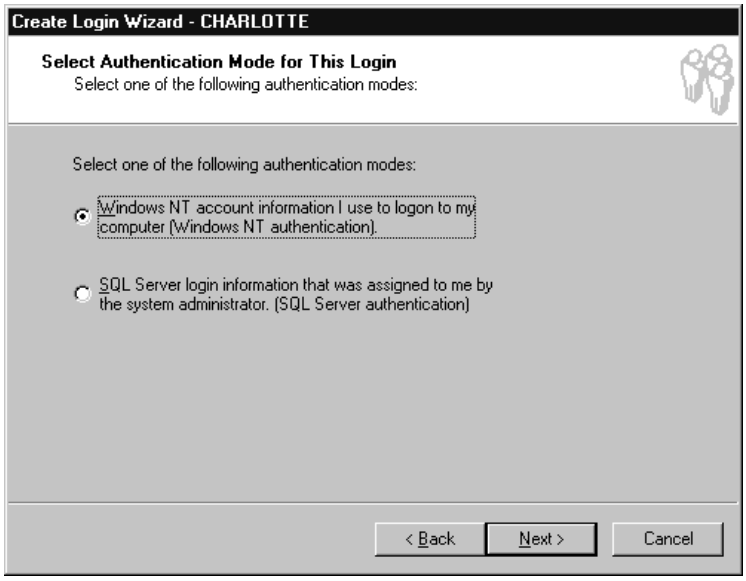


5. Select the Create Login Wizard choice under the Database node.

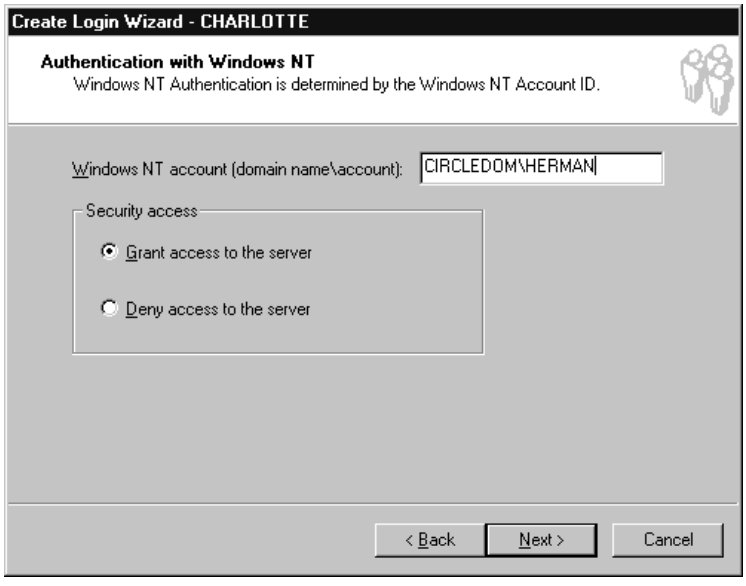


6. The intro screen for the Create Login Wizard appears.

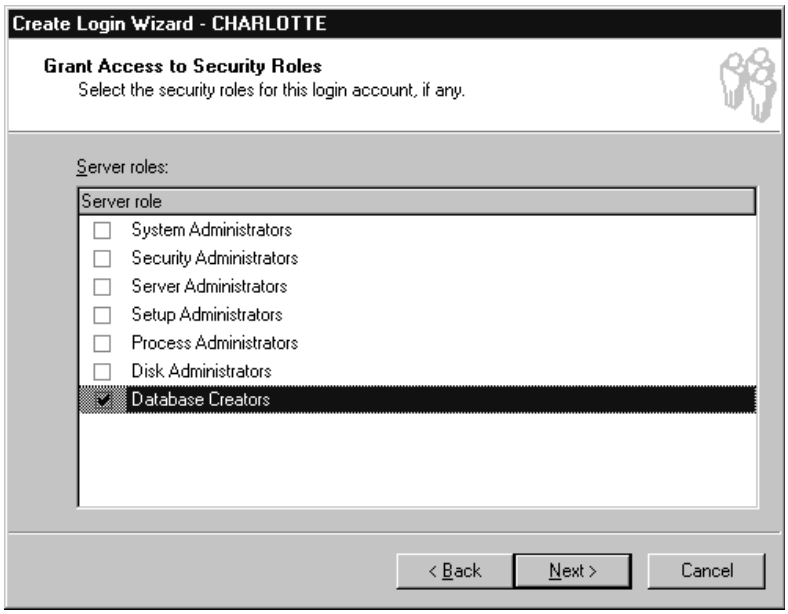
7. Choose the Windows NT Authentication option button.



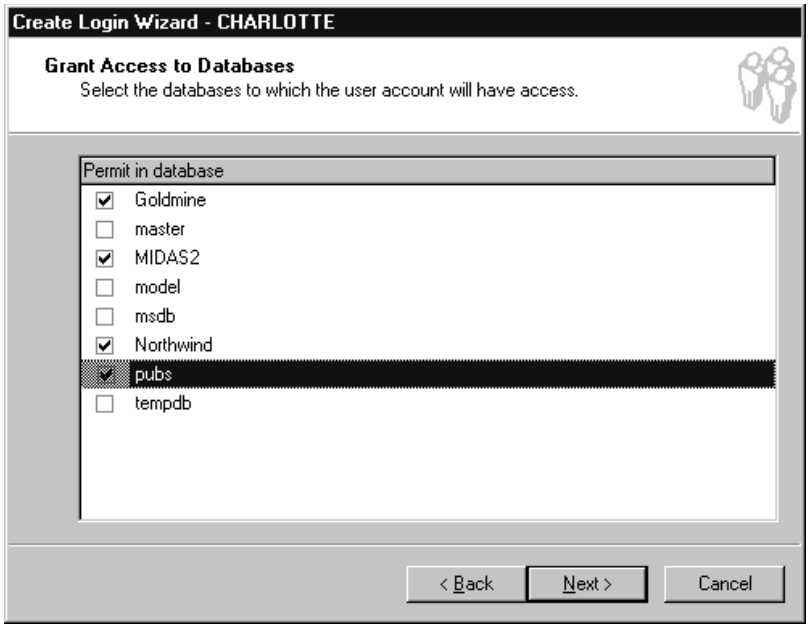
8. Enter the name of the domain and the user in that domain, and choose whether to grant or deny access.



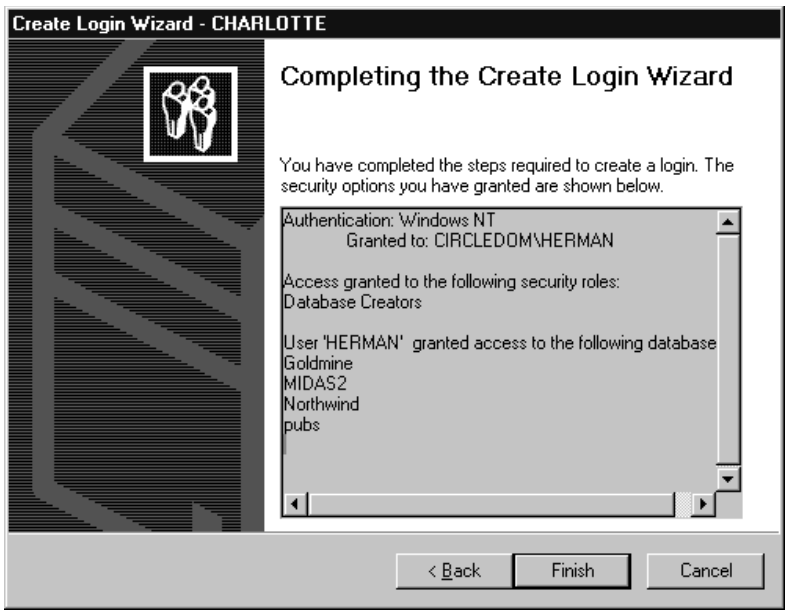
9. Select the security roles for this user.



10. Select which databases inside the SQL Server data store can be accessed by the user.



11. Click finish in the Completion screen.

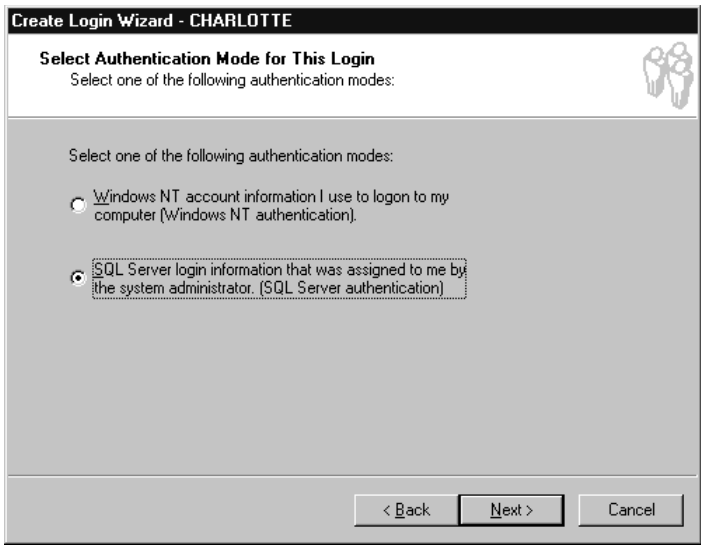


12. A confirmation dialog that indicates the user login was created displays.

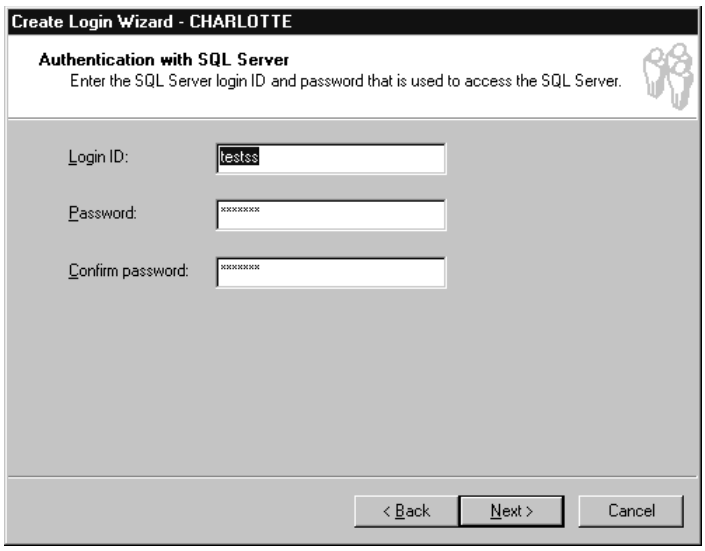
Creating a SS Authentication Login:

1. Follow steps 1 through 7 for NT Authentication

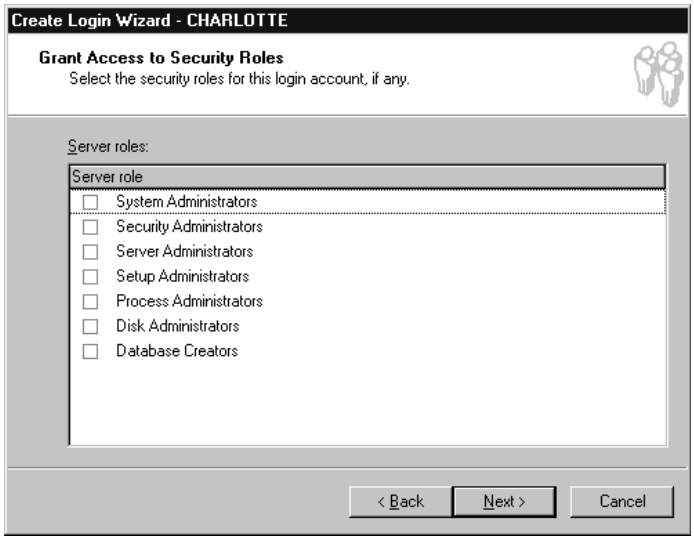
2. Choose the SQL Server Authentication option button.



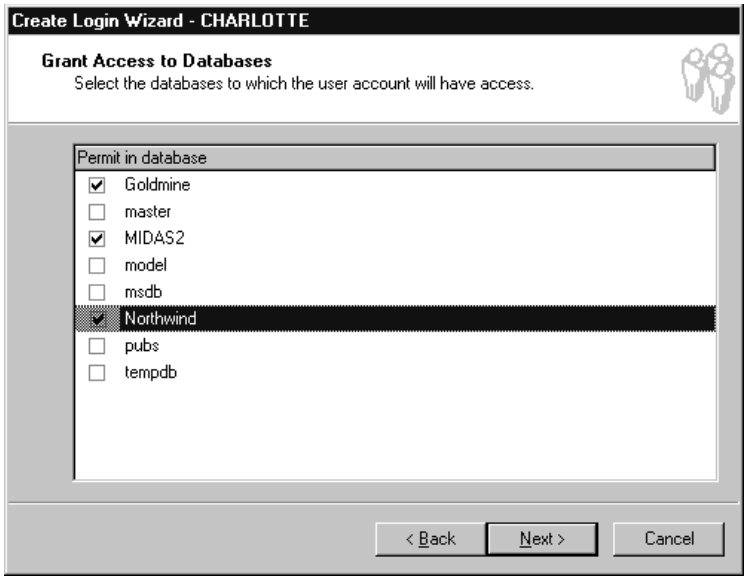
3. Create a Login ID, and enter and repeat the password



4. Select the security roles for this user.



5. Select which databases inside the SQL Server data store can be accessed by the user.

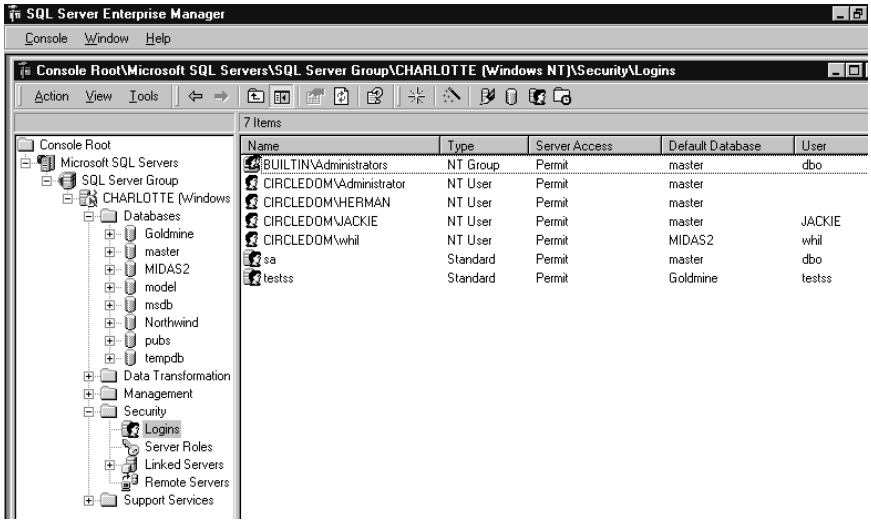


6. Click finish in the Completion screen.

You now see you've got both logins for both NT users and SS users in the SS:

Viewing logins (both NTA and SSA)

1. Open Enterprise Manager and drill down into Security, Logins



To log in to a SS database with the Query Analyzer via NT Auth

1. Open the Query Analyzer

2. Select the Connect menu option.

3. The following dialog will appear

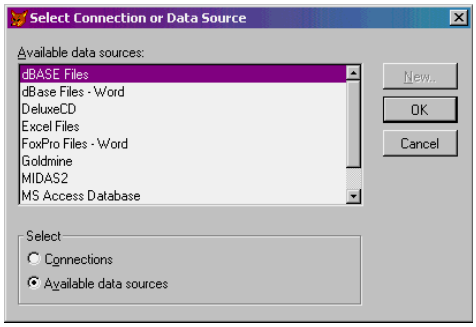
4. Select the Windows NT Auth option group, and click OK

The currently logged in user in Windows NT will be used authenticated in SS. If that user exists in SS, access will be granted. (Permissions, as said before, are another matter. See later.)

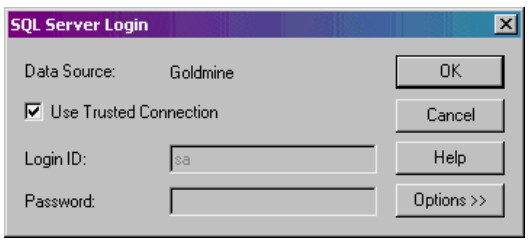
To log in to a SS database with a Remote View in Fox via NT Auth

You can create a Remote View in Visual FoxPro using NT Authentication through a data source.

1. Open a DBC
2. Create a Remote View
3. Select Available data sources
4. The Available data sources show Goldmine and MIDAS2 as available SQL Servers (these are ODBC DSNs that are stored in the client machine's Registry.)



5. Pick Goldmine, for example
6. The login dialog appears
7. Click the Use trusted connection (check box) – this means you want to use NT Authentication

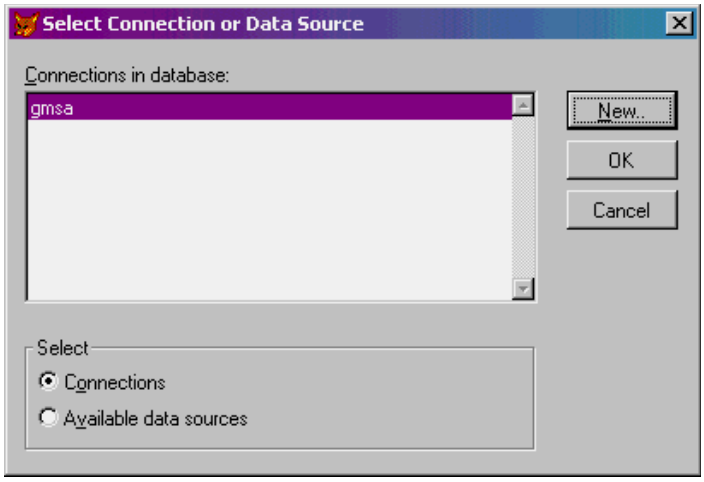


8. The View Designer will appear

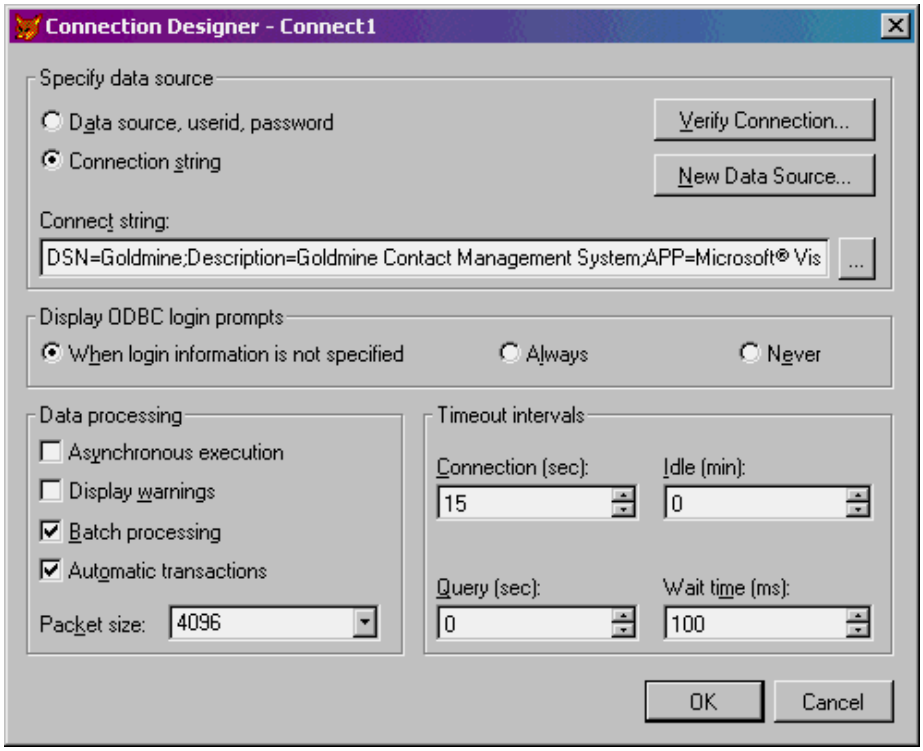
You can create a Remote View in Visual FoxPro using NT Authentication via a connection.

1. Open a DBC

2. Create a Remote View
3. Select the Connections option button
4. Click New



5. Get the Connection designer.



6. Type in the connection string if you know it or select New Data Source and drill down into the ODBC DSN that you need to build the Connection String. A connection string looks like this:

DSN=Goldmine;Description=Goldmine Contact Management System;APP=Microsoft® Visual FoxPro®;WSID=MACHNAME;DATABASE=Goldmine;Trusted_Connection=Yes

7. Note the last part – where "Trusted Connection=Yes" – this means that you're using NT Authentication.

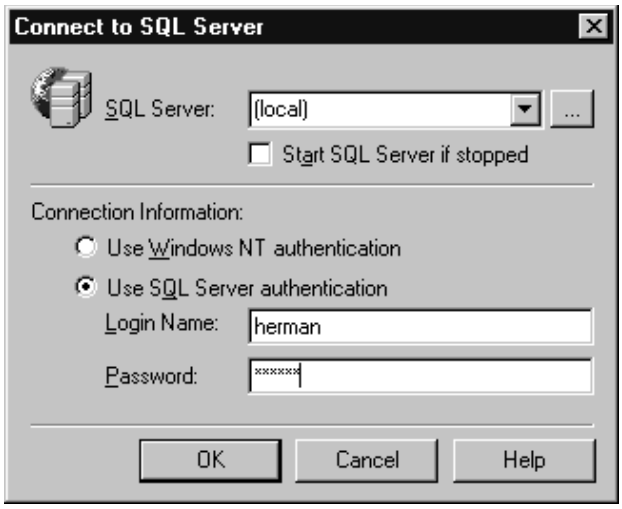
To log in to a SS database via SS Auth

1. Open the Query Analyzer

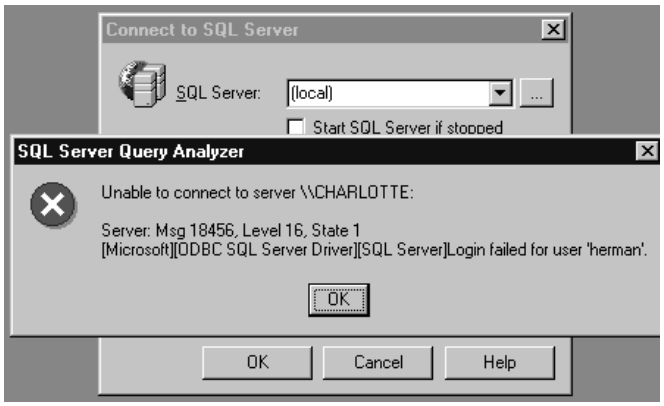
2. Select the Connect menu option.

3. The following dialog will appear

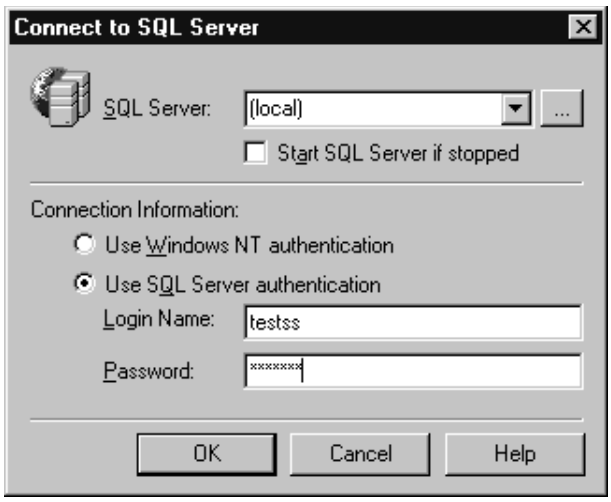
4. Enter the login id and password that you entered into the Create Login Wizard and click OK. Here, an incorrect user (a Windows user, not a SS login) is entered.



And you get an error



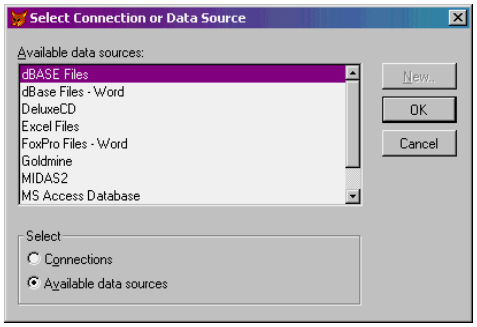
So enter the SS user, TestSS:



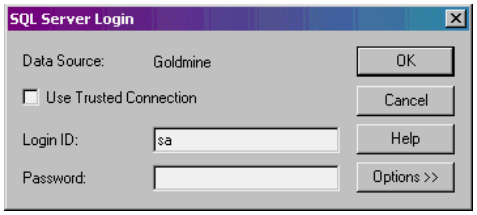
To log in to a SS database with a Remote View in Fox via SS Auth

You can create a Remote View in Visual FoxPro using SS Authentication through a data source.

1. Open a DBC
2. Create a Remote View
3. Select the Available data sources option button
4. The Available data sources show Goldmine and MIDAS2 as available SQL Servers (these are ODBC DSNs that are stored in the client machine's Registry.)

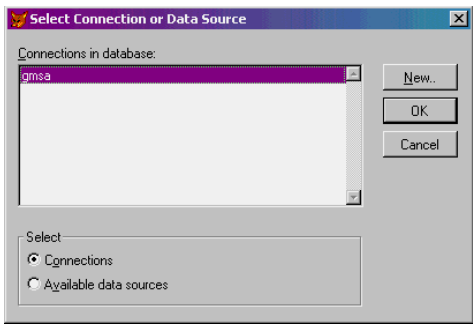


5. Pick Goldmine, for example
6. The login dialog appears
7. Enter SA/<blank>, or use testss / testbob for login ID and password



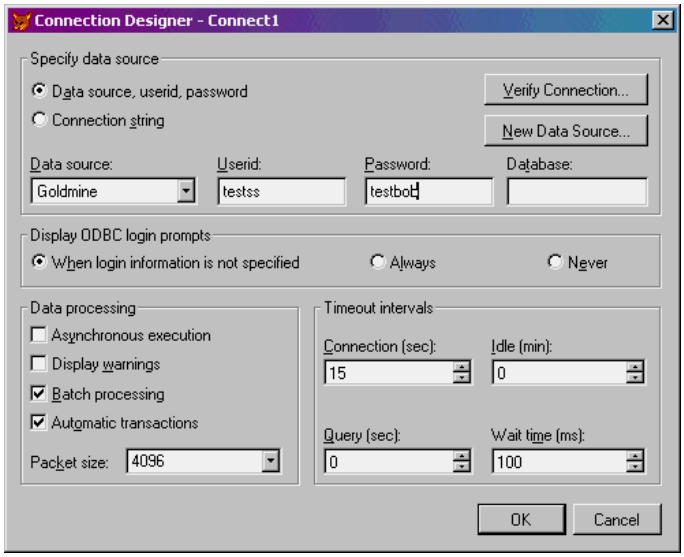
You can create a Remote View in Visual FoxPro using SS Authentication via a connection.

1. Open a DBC
2. Create a Remote View
3. Select the Connections option button
4. Click New

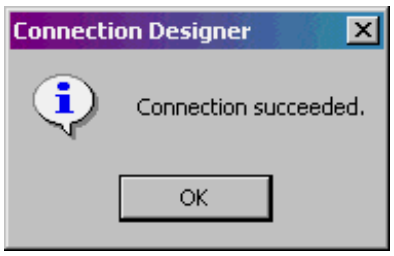


5. The Connection Designer opens

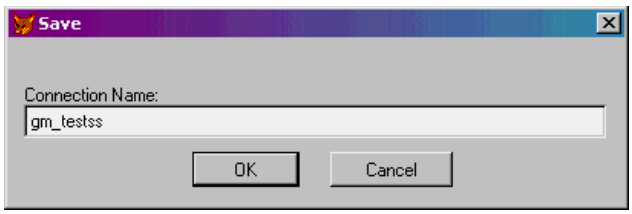
6. Enter a userid and a password that SS Authentication will recognize.



7. Test the connection with the Verify Connection button.



8. Finally, name the connection for use later.



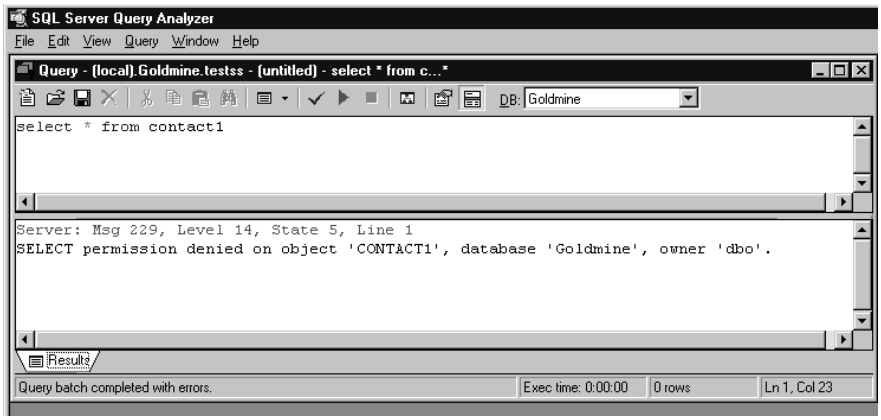
Permissions

To set a permission for a database user on a specific database:

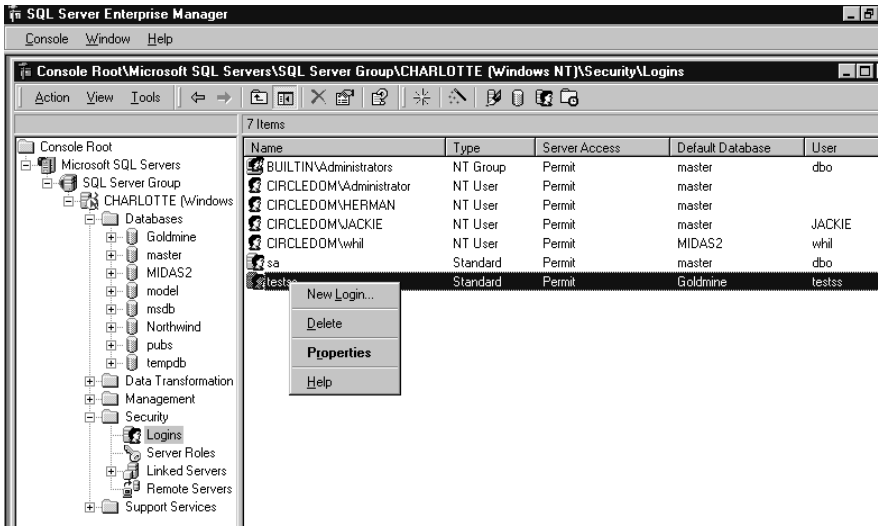
1. Open the Query Analyzer

2. Attempt to run a SELECT from a database where the user doesn't have the permission to do so:

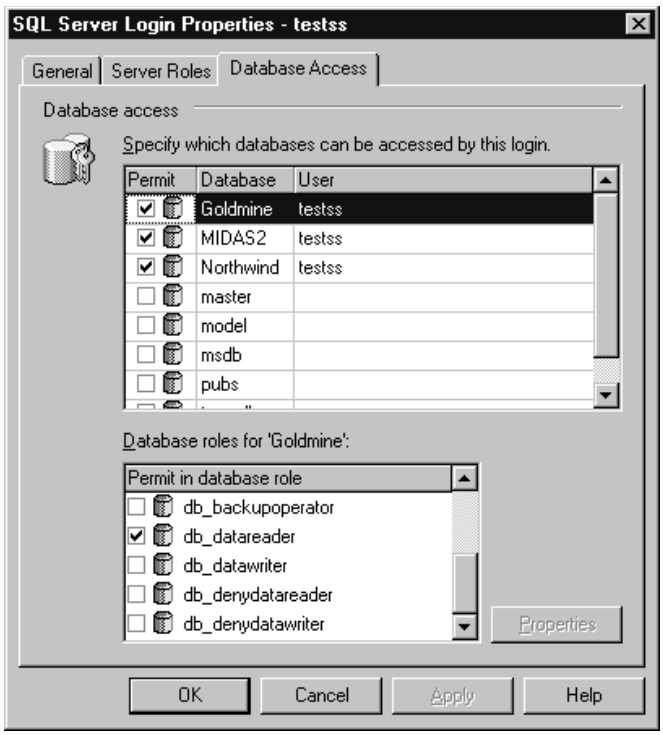
3. You'll get an error in the bottom half of the QA.



4. In order to set the appropriate permission – in this case, to allow the user to query the Goldmine database, right click on the user in the Security node of the Enterprise Manager



5. Select properties and click on the Database Access tab. Click on the datareader permission for the database of interest.


SQL Server Login Properties - testss
General
Server Roles
Database Access
Database access
Specify which databases can be accessed by this login.
Permit
Database
User
Goldmine
testss
MIDAS2
testss
Northwind
testss
master
model
msdb
pubs
Database roles for 'Goldmine':
Permit in database role
db_backupoperator
db_datareader
db_datawriter
db_denydatareader
db_denydatawriter
Properties
OK
Cancel
Apply
Help