Domain and simple

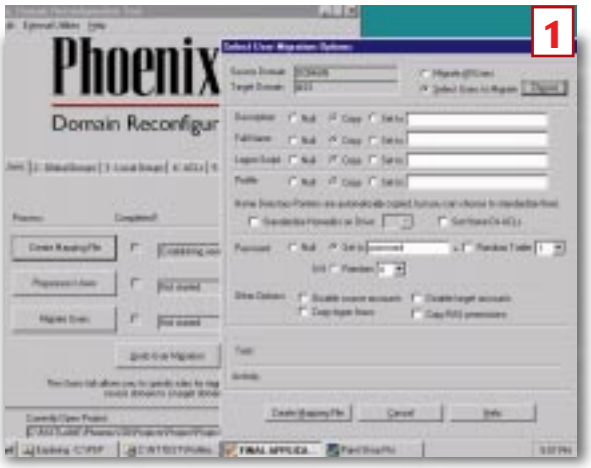
Companies often end up with a proliferation of domains, and trying to rationalise the structure can be a daunting task. Bob Walder has ways of making it easier.

Much has been written about the lack of a true directory service in NT and the problems inherent in managing multiple domains in an NT network. But if domains are that complex to administer, why do organisations so often finish up with tens or hundreds of the things spread about the place?

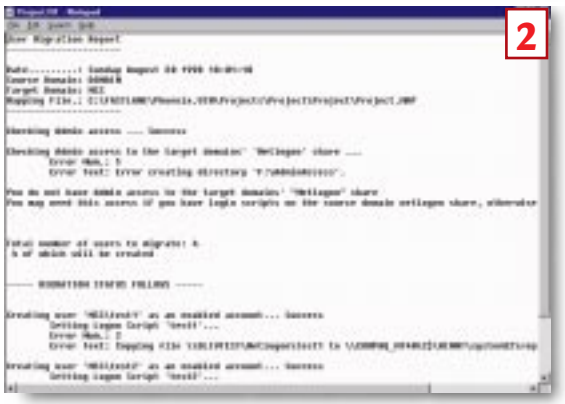
History lesson

There are a couple of historic reasons for this. One is that no matter what people tell you, size does matter. In Windows NT Server 3.1, domain controllers were limited to storing 10,000 objects in the security accounts manager (SAM) database. Many larger companies found this to be insufficient as the network grew and were forced into using multiple domains. With the release of Windows NT Server 4.0, the limit was increased to 40,000 users and the maximum recommended size of the database was 40Mb. But for some it was too late since the domain structure was already fixed, and for others even this number remained too small.

There are other reasons, too, of course. Sometimes a network's communications infrastructure dictates the domain structure to minimise replication across slow WAN links. Large organisations might also want to delegate administrative tasks to a number of people and the only true security boundary in an NT network is the domain. Finally, there is simple growth, whether organically within a company or via acquisition of others. Either way, it is possible to finish up with numerous domains which would be better merged together. The problem is that as things change, it is often necessary to collapse the domain structure to something simpler. When two companies are merged, for instance, rationalisation of the domain structure is always desirable. Microsoft recommends collapsing domains in preparation for the move to NT5 and Active Directory. Unfortunately, domain reconfiguration is not that straightforward because of the unique security identifier (SID) associated with each user object.



1

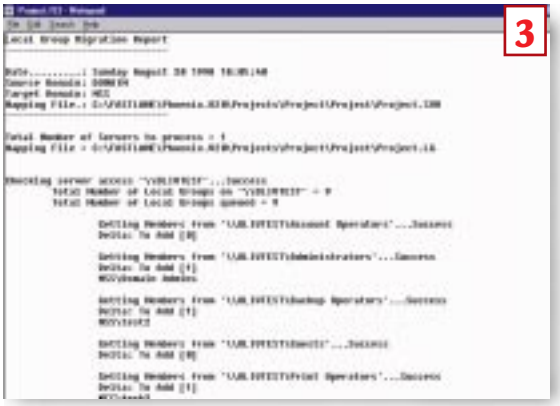

2

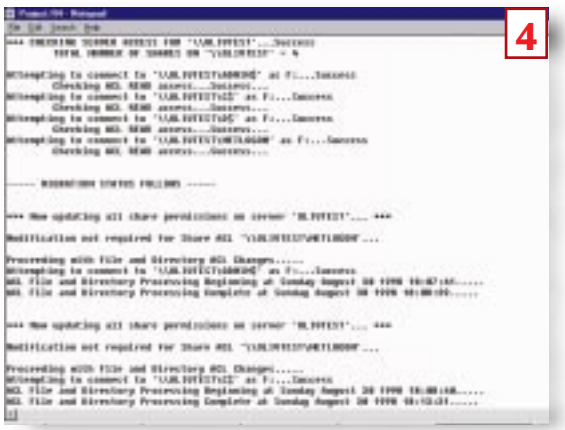
It is an unfortunate fact of life in the NT world that SIDs are not portable across domains. And security is completely dependent on the SID rather than the user name. It is the SID that is a member of a Local Group and ACL and User Rights, *not* the user name, which is there for display purposes and ease-of-management only. To merge two domains into one, therefore, would require manual creation of users and groups in the target domain, and manual adjustment of the various shares and access rights to ensure that these users could continue to access resources in the source domain until the migration was completed.

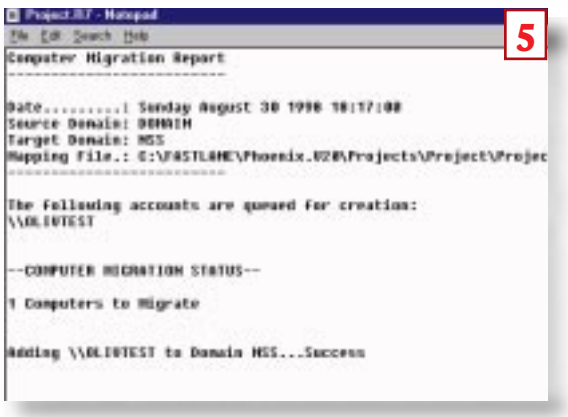
Rationalisation of the domain structure is always desirable

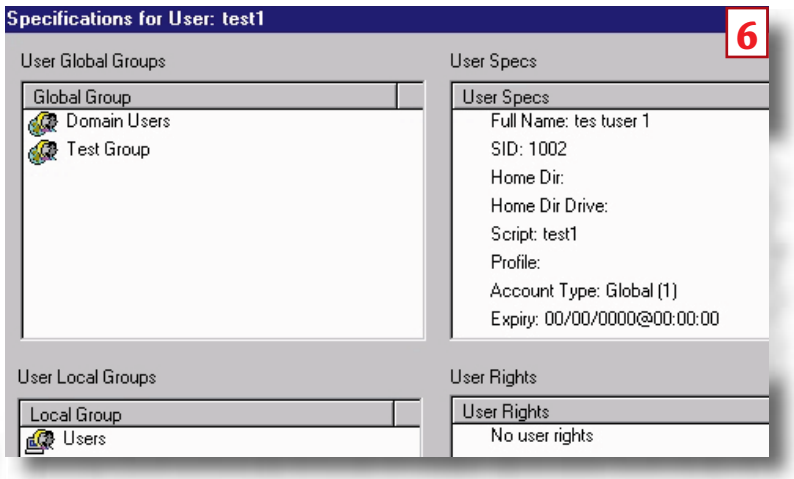
Domain reconfiguration in a large enterprise cannot be accomplished overnight, so you have to ensure that as users are moved to the new domain, they still have access to all their original shares, mailboxes, printers and other resources until the migration is finished. I recently came across a product called Phoenix, from FastLane Technologies, which helps with the whole reconfiguration process. Going through the various stages step by step I will highlight the problems and show where Phoenix may be useful. In a large network with many domains, these stages represent a huge manual effort which is not only tedious but prone to errors. Tools like Phoenix, which should probably have been included as part of the base operating system, certainly take much of the pain out of the process.

1 **Users** [Fig 1]. Assuming we already have NT Server installed on a primary domain controller in our new domain, the first task is to create the user objects that will be members of the new domain. Where we are consolidating a number of domains into one, this can be a huge task. Phoenix allows the administrator to perform this creation automatically, selecting the users from a list of those









already on the existing domains. User names to be migrated can be chosen individually from a pick list, or can all be migrated in one go with the appropriate user objects duplicated (with new SIDs) in the target domain. The remaining stages focus on finding an objects source SID and inserting the SID of the destination object so that the new users maintain the same access as the original account.

2 Global Groups [Fig 2]. Like users, global groups cannot be copied across domains so new objects need to be created in the target domain to mirror the source groups, and the appropriate users must be added to the new groups. Once again, this is a tedious manual process that can be automated using Phoenix. If multiple domains are merged to one target, groups with the same name can be merged or created as separate groups. For example, you probably wouldn't want to merge all the "Domain Admin" groups into a single object in the target domain. As the new global group objects are created, they are populated with the same members that were in the source groups, using the new user objects created in the target domain in stage 1.

3 Local Groups [Fig 3]. In order that a newly-created user object has the same access to resources as the original account, Phoenix searches all local groups for the original users' SIDs and appends the SID of the target user account. Global groups in the source domain are processed in the same way, since local groups can include both users *and* global groups. As this process can be resource intensive, you can run the external application locally on each server throughout the enterprise. This can be done using the DR Distributor, which distributes a secure scheduler and an automated updater down to local computer level. The administrator can then push the updating of local group migration, ACLs and user rights to the computer itself to minimise the load on the network. The local computer spawns the update application and updates its own data locally, as a central console maintains and reports all updates.

If an organisation is implementing a staggered migration, it is possible to schedule the automated updater on the remote computers to run every day, which allows the network to quickly adapt to massive changes in stages.

4 ACLs [Fig 4]. Like local groups, all ACLs must be searched for SIDs of the original account. As each is located, the SID of the target account is appended, therefore ensuring that the target user object retains the same access rights as the original.

5 Rights [Fig 5]. Although the previous stages give the destination user access to the same global groups, local groups, shares, files and directories, the user still may be unable to log on. Such computer-specific access is determined by user rights and advanced user rights, and this stage ensures that once again destination SIDs are appended wherever source SIDs are found, to ensure that the new account has equal access to the same physical computers as the old account.

6 Computers [Fig 6]. The final stage of domain reconfiguration creates computer accounts on the target domain for all computers in the source domain. Other tools in the Phoenix package make light work of moving Domain Controllers and Exchange mailboxes between domains, too. Once the initial migration phase has been completed, Phoenix can also be used as a day-to-day domain management tool.

PCW CONTACTS

Bob Walder *can be contacted via the usual PCW editorial office (address, p10) or email* networks@pcw.vnu.co.uk

FastLane's Phoenix *is available from Peapod Distribution on 0181 606 9990*