DB2 Universal Database (UDB) and Windows Security

1 Introduction

Security is one of the most important features of any database management system. It is key to keeping data safe and limiting data access to those with a need to know. DB2 UDB has many security features of its own, but on Windows NT it also relies on the security features of the Windows NT operating system itself. To understand security for DB2 UDB on Windows NT, we must understand not only the many security features of DB2 UDB but also the Windows NT security model and how it's used by DB2 UDB.

Let's take a moment to look first at the security issues that must be resolved. What exactly are the problems we're solving by implementing security features?

First, providing security for a database management system means providing a means of authentication. Authentication means "are you who you say you are?". This must be answered before you have any access at all to the database management system. The Windows NT security system may be involved in providing this authentication. Authentication can take place at any one of a number of levels: locally at the client, at the DB2 server, at the domain level, through a trusted domain. We'll look at all these scenarios as we discuss the Windows NT security model.

Second, once you have been authenticated and DB2 UDB knows who you are, what can you do? You may need to protect data from some users but not others. You may need to give some users the ability to perform administrative functions on the database system but not see the data. Control over what any individual user can do is maintained through privileges. We'll discuss these in depth as we discuss DB2 UDB's security features.

Before we proceed, a few definitions, listed below in Table 1, will help to clarify some of the terms we use throughout this paper.

Table 1

Terms	Definition (for this book)
Windows NT	Either Windows NT Workstation or Windows NT Server. Reference will be made to the common Windows NT features of the two products.
Windows NT Workstation	The Windows NT Workstation product. Cannot be a domain controller.
Windows NT Server	The Windows NT Server product. It is a superset of the NT Workstation product. A machine running Windows NT Server may be a Windows NT Workstation or a domain controller.
Domain	A domain is an arrangement of client and server computers referenced by a specific (unique) name that share a single user accounts database.
Domain Controller	Refers to the computer running Windows NT server that manages all aspects of user-domain interactions and uses the information in the domain user accounts database to authenticate users logging onto domain accounts.
Primary Domain Controller	In a Windows NT server domain, the computer running the Windows NT server that authenticates domain logons and maintains the user accounts database for the domain. There

	can be only one per domain. It can also be referred to as the PDC.
Trust Relationship	Trust relationships are an administration and communications link between two domains. A trust relationship between two domains enables user accounts and global groups to be used in a domain other than the domain where these accounts are actually defined. Domains use established trust relationships to share account information and validate the rights and permissions of users and global groups residing in the trusted domain. Trusts therefore simplify administration by combining two or more domains into a single administrative unit.
Backup Domain Controller	These are servers that contain up-to-date and accurate copies of the user accounts database. These servers can also authenticate workstations in the absence of a primary domain controller (PDC). They can also be referred to as BDCs.
Server	A Windows NT Server that is part of a domain as a file, print or application server, but is not a domain controller.
Workstation	A machine running Windows NT Workstation or Windows NT Server in a domain that is not a domain controller or a file, print , or application server.
Right	The ability of a user or group of users to perform a Windows NT operation. Examples of rights are logging on to a server and performing backups. Rights apply to the computer as a whole, as opposed to permissions, which apply to specific objects. These can also be termed *user rights*.
Permission	Authority in Windows NT granted to a user or group of users to perform operations on specific objects, such as files, directories, printers, and other resources. Examples of permissions are read, change, full control and no access. Permissions are applied on a user-by-user or group-by-group basis.
Instance	DB2 Adminstration unit. On a Server it represents an independent set of databases. A DB2 Server Instance runs as an NT Service.
Privilege	Within DB2 UDB, the right of a particular user or group of users to create, access or modify an object.
Windows 2000	Either Windows 2000 Professional or Windows 2000 Server. Reference will be made to the common Windows 2000 features of the two products.
Windows 2000 Workstation	A machine running Windows 2000 Professional. Cannot be a domain controller.
Windows 2000 Server	A machine running Windows 2000 Server, Windows 2000 Advanced Server or Windows 2000 DataCenter Server. A machine running Windows 2000 Server may be a member server or a domain controller.
Windows 2000 Domain Controller	Any machine running Windows 2000 Server can be configured to be a domain controller.

List of Definitions

2 Windows NT Security Concepts

In this section we'll look at some of the Windows NT concepts that are key to understanding how DB2 UDB utilizes the Windows NT security model.

Windows NT has three models for creating a logical organization of computers: the workgroup, the client-server and the domain model. For our purposes we will discuss only the workgroup and the domain models.

Both workgroups and domains provide ways to share resources such as file and printing services. They do have some significant differences however, the primary one being security.

2.1 Workgroups in Windows NT

A workgroup is most simply described as a collection of Windows workstations. It can contain computers running any number of operating systems including DOS, Windows 3.x, Windows for Workgroups, Windows 95/98, Windows NT Workstation and Windows NT Server, and is identified by a unique name.

Windows NT Server (installed as a stand-alone/member server) or Windows NT Workstation installations in the workgroup can share their objects with other members (clients) of the workgroup. These shared resources, or shares, might include directories or printers. A share will have a share name associated with it. Although these resources are shared, user logons are specific to each computer in the workgroup. Therefore, if a user requires shared access to all servers in a workgroup, an account must be created for that user on each machine.

Although the workgroup model can work well in a small organization where centralized security is not required, security can be a concern in this environment. Shared objects on a network can be made more secure by either hiding the share or by putting a password on them. Hiding a share means that it will not appear on browse lists on other machines in the workgroup and has to be known by anyone who wants to use it.

2.2 Domains in Windows NT

A Windows NT **domain** (hereinafter referred to as a domain) is an arrangement of client and server computers referenced by a specific (unique) name that share a single user accounts database. The key difference between a Windows NT workgroup and a Windows NT domain is that the computers that make up the domain share a centralized user accounts database. This is an important concept which differentiates the implementation of DB2 UDB on Windows NT from its implementation on all other platforms.

Unlike the model for workgroup security the userid and password only need to be defined at the Primary Domain Controller to be able to access domain resources. In contrast, the Windows NT workgroup environment would require the userid and password to be defined on each machine the user wanted to access.

2.2.1 Primary Domain Controllers

A domain is established when a Windows NT Server is defined as a domain controller in a new domain. This can be done when a Windows NT Server is installed. During the setup procedure

you may select to either create a primary domain controller in a new domain, a backup domain controller in a known domain, or a stand-alone server in a known domain. Selecting controller in a new domain will make that server the Primary Domain Controller (PDC). A domain name must be supplied. The domain name must not be the same as any other domain on your network or any machine name within another domain. Each domain contains only one PDC.

As the PDC, a Windows NT Server will have the master copy of the **Security Access Manager (SAM)** database. The SAM database contains information about which users can log onto the domain, their passwords and which groups they belong to. It also records what machine names are members of the domain and what other domains this domain knows about and trusts. All other servers added to the domain as Backup Domain Controllers (BDC) will hold a copy of the SAM database which is regularly synchronized against the master copy on the PDC.

2.2.2 Backup Domain Controller

Once a domain has been established (a Primary Domain Controller was created), other Windows NT Servers can be added to the domain as either normal servers or as domain controllers. If added as another domain controller, a Windows NT Server is called a Backup Domain Controller.

The major difference between a PDC and a BDC is the implementation of the SAM database. A BDC holds a copy of the SAM database from the PDC. The BDC can authenticate users to the domain on behalf of the PDC.

The main purpose of a BDC is to provide redundency in the Domain should the PDC machine fail. If a PDC fails, a BDC can be promoted to be a PDC.

2.2.3 Windows NT Server and Windows NT Workstation

When a Windows NT Server is installed on a domain not as a domain controller, it can operate as a file, print or application server. The workstation with Windows NT Server installed behaves much like a system with Windows NT Workstation installed.

A computer running Windows NT Workstation or Windows NT Server can belong to either a workgroup, a domain or neither (stand-alone), but cannot belong to both a workgroup and a domain.

A Windows NT Workstation machine will have a local SAM database as well. It can authenticate users logging in locally and decide what rights users have when logging on from that machine. This Windows NT machine is not capable of becoming a Backup or Primary Domain Controller. To become a domain controller, Windows NT Server must be installed.

2.3 Groups and User Authentication

Users are defined on Windows NT by creating user accounts using the Windows NT administrative tool called the User Manager. An account which contains other accounts, called members, is a **group.** Groups give Windows NT administrators the ability to grant rights and permissions to a number of users at once, without having to maintain each user individually. Groups, like user accounts, are defined and maintained in the SAM database of Windows NT machines.There are two types of groups in the Windows NT architecture:

- **Local groups**
- **Global groups**

A **local group** can include user accounts that have been created in the local accounts database. If the machine is part of a domain, it can also contain domain accounts and groups from the Windows NT domain.

A **global group** exists only on a domain controller and contains user accounts from the domain's SAM. A global group can be used in servers and workstations of its own domain, and in trusting domains. It can become a member of local groups on those machines. Rights cannot be granted to Global groups. Their only use is for inclusion in local groups.
The Primary Domain Controller holds the SAM for the domain. This SAM is replicated to any BDCs[1] in the domain. Domain controllers do not have a *local* SAM database. They hold user and group data for the domain. In this sense, any groups created on the PDC, local or global, are domain groups.

Windows NT machines that are not domain controllers (NT Workstations and some NT Servers) will each have their own SAM databases. User accounts and groups created on those machines are local to that machine. There is no Create Global Group option on machines that are not domain controllers.

2.3.1 Global Groups

A global group can only contain user accounts from the domain on which it is created. It cannot contain any other groups as members. However, once created, global groups can be seen and used by any machine in the domain or machines in trusting domains.
Windows NT Server comes with a number of default global groups. Those groups are:

- **Domain Users**
- **Domain Admins**
- **Domain Guests**

The Domain Users group contains all user accounts created on the domain.
The Domain Admins group contains designated administrator accounts. By default, Domain Admins contains only the Windows NT default administrator account called Administrator.
The Domain Guests group contains all guest accounts for the domain. By default, Domain Guests contains only the default guest user account called Guest.

2.3.2 Local Groups

Local groups are local to the Windows NT machine on which they are created. Remember that group information is stored in a machine's SAM database. A local group created on a workstation is specific to that workstation. A local group created on a domain controller, however, applies to all domain controllers in that domain because the SAM is propagated to all domain controllers on a regular basis.

Local groups can contain individual user accounts which are defined in the local SAM database, any users from within the domain or users from trusted domains. In addition a local group can contain as a member a global group from the domain or a trusted domain.
A local group cannot contain other local groups. Windows NT has a number of default local groups established at installation. The default local groups are:

- **Account Operators (Windows NT Server only)**
- **Administrators**

[1] From the Administrators point of view the SAM on a BDC is read-only. All User Management update activity must be performed at the Primary Domain Controller.

- **Backup Operators**
- **Guests**
- **Power Users (Windows NT Workstation only)**
- **Print Operators (Windows NT Server only)**
- **Replicator**
- **Server Operators (Windows NT Server only)**
- **Users**

There is one additional group called Everyone. This group does not appear on the list of groups in User Manager. However, it can be assigned rights and permissions. Anyone who has a user account in the domain, including all local and remote users, is a part of the Everyone local group. The Everyone group also contains all global groups of any trusted domains.
The Administrators group is the most powerful group. By default, it contains only the default administrator account called Administrator. The Administrators local group on the Primary Domain Controller is the Administrators group for all domain controllers of that domain.

When a Windows NT Workstation or Windows NT Server joins a domain (but not as a domain controller, in the case of Windows NT Server), the Domain Administrators global group is added to the Administrators local group on the machine. This gives any member of the Domain Administrators group administrative privileges on that machine.

A user ID must be known to at least one local group on a Windows NT machine before that user is allowed to log on at that machine.

Members of the Users group have minimal rights on machines running Windows NT Server, but they do have rights on Windows NT Workstations. They have the right to manage and create local groups. When a Windows NT machine joins a domain (not as a domain controller, in the case of Windows NT Server), the Domain Users global group is added to the Users local group on the machine. This allows all domain users to log on to the domain from that machine. The Logon Dialog will provide a drop-down list box that will provide the option of logging on to the local machine, the domain or any domain trusted by the domain that machine is a member of. See the discussion below about establishing Trust Relationships between Domains.

2.4 Windows NT Authentication

We have discussed the concepts of Windows NT user accounts, local and global groups and domains. The next logical topic is authentication. The actual process of user authentication is relatively simple. Authentication is verifying that a user is who they say they are.

Recall that user IDs and passwords are stored in the SAM database on Windows NT machines, but a user's user ID and password does not necessarily have to reside on the machine from which they log on. When Windows NT authenticates a user, it follows a simple hierarchy to look for a user ID and password. If you choose a workstation or local logon, Windows NT will only look at the local SAM. If the user is not in the local SAM, authentication will fail.

If you choose domain authentication the domain controller that does the authentication can be either the PDC or a BDC. BDCs have a copy of the PDC's SAM database. To determine which domain controller will perform the authentication, a broadcast message is sent out from the user's machine, and the first domain controller to respond to the message will perform the authentication.

If the user is not known to the domain, (that is, the user ID is not in the SAM database of the PDC), then domain controllers of any trusted domains are queried. Either the PDC or a BDC can respond to an authentication request from a trusting domain.

Once the userid has been found and the password authenticated, any account or policy restrictions are determined, as well as a list of groups of which the user is a member.

2.5 Trust Relationships Between Domains

We have discussed the concept of a single domain; however, an enterprise may wish to establish more than one domain. These domains do not have to exist independently, nor do separate user accounts have to exist for each domain a given user wishes to log into. Interdependent multiple domains can be achieved through relationships between domains called trusts.

2.5.1 Trusted Domains

Trust relationships between domains are established so that users from one domain can access resources in another domain without being re-authenticated.

There are two characteristics of a trust relationship:

1. One domain trusts another to authenticate users on its behalf and therefore grants access to resources in its domain without re-authenticating users.

2. An administrator from one domain trusts an administrator from another domain to administer resources in that domain.

The two domains in a trust relationship are called the **trusting** and the **trusted** domain. A trust relationship lets an administrator of one domain (the trusting domain) grant rights and permissions to global groups and users of another domain (the trusted domain). The administrator of the trusted domain must be, in turn, trusted since this administrator can control which users are members of global groups.

Trust relationships are not transitive. This means that explicit trust relationships need to be established in each direction between domains. There is no concept of an implicit or piggybacked trust relationship.

2.6 Models of Domain Trust

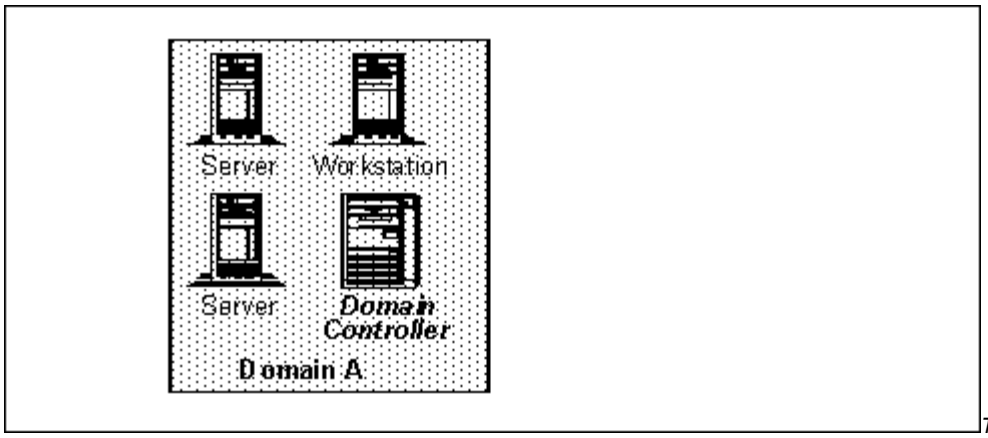
Choosing the right domain trust architecture for an enterprise can be an involved and complex task, with a number of considerations to be taken into account. To assist in this process, let's look at four common models of domain organization. They are:

1. **The Single Domain Model**
2. **The Master Domain Model**
3. **The Multiple Master Domain Model**
4. **The Complete Trust Model**

These should be treated as models. Organizations should configure their domain(s) to best suit their individual needs. We will examine each of these models briefly and consider how the model might be used with DB2. Other important factors will influence the model an organization chooses. Consult a Windows NT planning reference for more information.

The Single Domain Model

Figure 2 represents the single domain model.



The Single Domain Model

All servers and workstations belong to one domain. There are no trust relationships to any other domain.Advantages of this model include:

- **It's easy to implement.**
- **It's a suitable design for a small to medium sized network**.
- **There are no trust relationships to establish or maintain.**
- **You have one set of administrators.**

Disadvantages of this model include:

- **The list of users and machines can grow to an undesirable size.**
- **Network and server performance problems may arise.**

An example of a single domain model might be a small network with an independent domain. This could be a production environment, where it is desirable to keep the production data separate from the development environment. You might also have a number of small domains for an organization where the sharing or dividing of resources such as databases is not required. The ability to administer each domain separately is not an issue.

The most compelling reason not to implement this model, especially in a production environment, is generally the size of the domain, specifically the number of users and machines. These factors affect the size of the SAM database on the domain controllers.

The Master Domain Model

The following figure illustrates the **master domain** model:

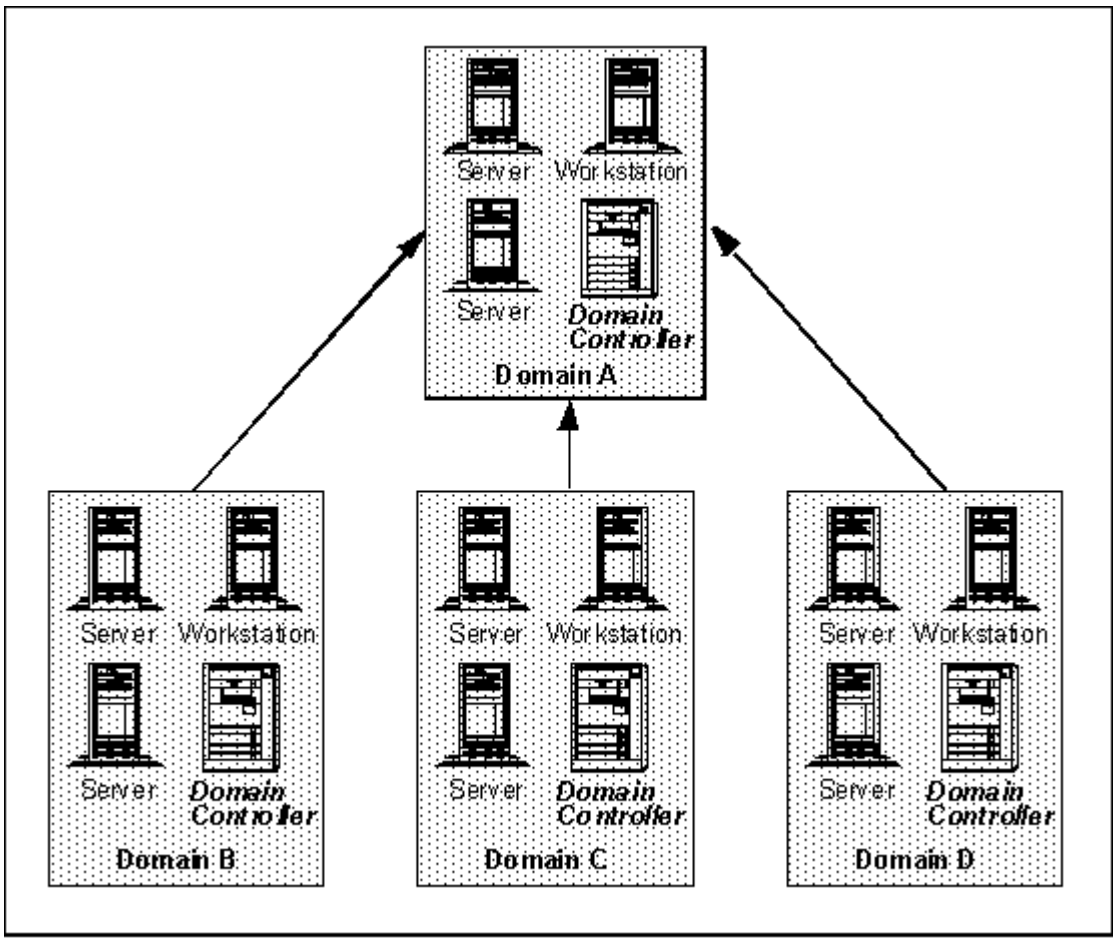


Figure 2. *The Master Domain Model*

The master domain, Domain A in the illustration, is the domain where all users are defined. All other domains trust the master domain. Domains other than the master domain have no users defined. The other domains are called **resource** or **slave domains**.

Advantages of this model include:

- **Administration for the enterprise is centralized.**
- **It supports logical grouping of resources (such as divisions or departments).**
- **It supports geographical division of an enterprise.**
- **Global groups are defined only once.**

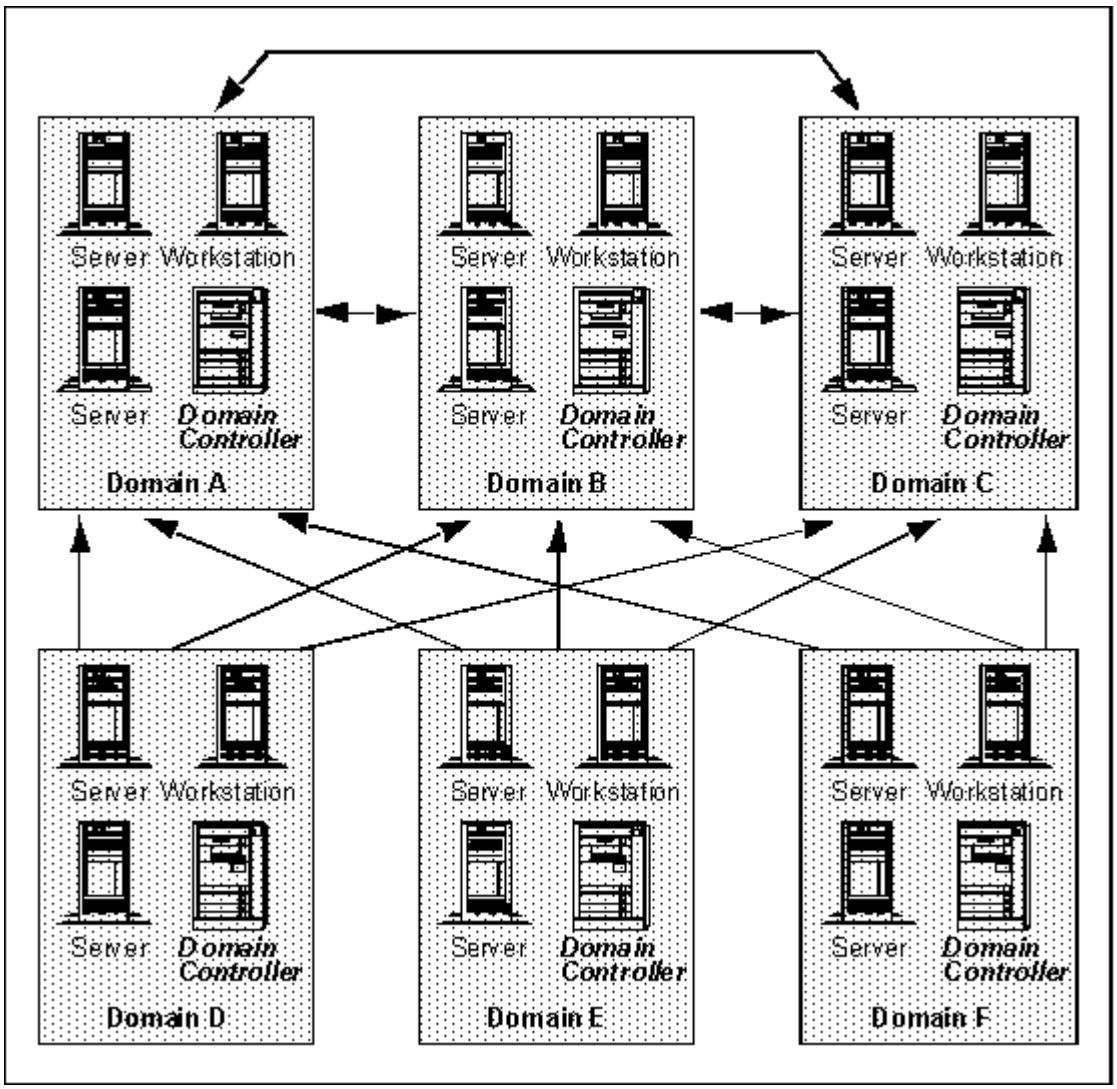
Disadvantages of this model include:

- **Performance may degrade on a WAN or with a large number of users.**
- **Local groups must be defined on each domain.**
- **Global administration can be cumbersome to establish.**
- **Master domain can be a single point of failure.**

You might find the master domain model implemented where each department in an organization is on its own domain. However, all administration and authentication occurs in the master domain. The enterprise is split geographically and resources are grouped accordingly. However, users are all defined and administered centrally. This model easily supports movement of personnel across domains.

The Multiple Master Domain Model

The following figure shows the **multiple master domain** model.



The Multiple Master Domain Model

In this example, Domains A, B, and C are master domains where the user accounts are held. Domains D, E, and F are resource domains. The master domains have trust relationships between each other and can each authenticate for the resource domains.

Advantages of this model include:

- **It supports a large number of users with acceptable performance.**

- **Resources are grouped logically.**
- **Resource domains can be managed independently for security**.

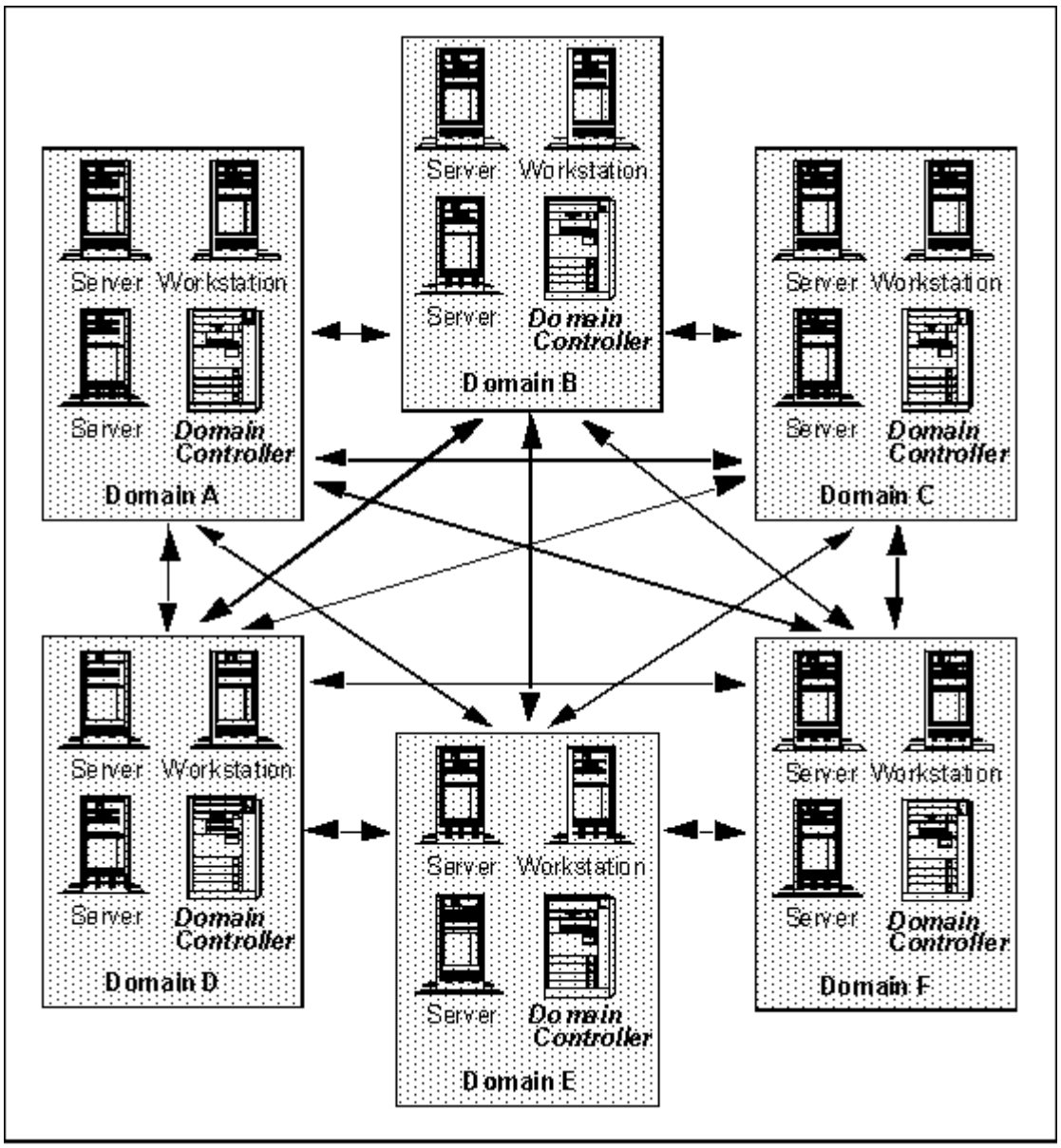
Disadvantages of this model include:

- **Groups may need to be defined more than once for different domains.**
- **There are many trust relationships to manage.**
- **Maintenance of user accounts is more difficult because they are in multiple domains.**

A multiple master domain may be established for the same reasons as a master domain. You may choose the multiple master model if you have too many users for one domain to handle all the authentication requests. To ease network traffic and speed user authentication requests, multiple master domains are created to service the resource domains.

The Complete Trust Model

The following figure shows the complete trust model.



The Complete Trust Model

In a complete trust model, domains exist with trust relationships to and from all other domains on the network.

Advantages of this model include:

- **It supports a large number of users.**
- **It does not require central administration.**
- **Resources and users are grouped logically into domains (from a browser perspective).**
- **Resources are managed independently for each domain.**

Disadvantages of this model include:

- **Lack of central administration can cause potentially severe network problems**

- **There are a large number of trust relationships to manage.**

An example of a suitable environment where the complete trust model might be implemented is a development environment.

3 Windows 2000 Security Concepts

In this section, we describe some of the Windows 2000 concepts that are different from Windows NT. While the basic concepts of users and groups (local, domain local, and domain global group) have not changed, the organization of domains into domain trees and forests is new to Windows 2000.

In Windows 2000, the security service is integrated with Active Directory through logon authentication and access control to objects in the directory.

Windows 2000 supports Kerberos V5 authentication in addition to the Windows NT Lan Manager (NTLM) authentication.

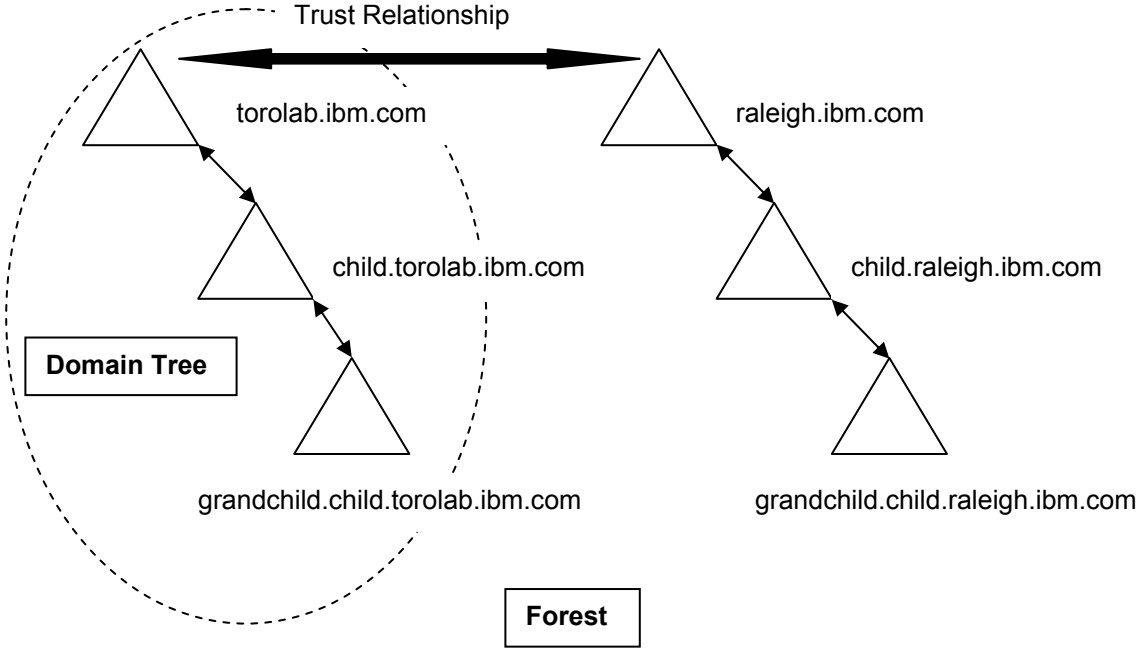
3.1 Windows 2000 Domain Controllers

A domain controller is a computer running Windows 2000 Server that has been configured using the Active Directory Installation wizard. A domain can have one or more domain controllers. A typical small organization may need only one domain with two domain controllers for high availability and fault tolerance. A large corporation will need one or more domain controllers in each location to provide high availability and fault tolerance.

Windows 2000 supports multimaster replication of data between domain controllers. Therefore, each Windows 2000 domain controller can accept change and replicate the changes to the other domain controllers in the domain. This is an enhancement over the Windows NT model in which only one server in the domain, the Primary Domain Controller, can perform update operations.

3.2 Domain Trees and Forests

Each domain in the directory is identified by a DNS domain name and requires one or more domain controllers. If your network requires more than one domain, you can easily create multiple domains.



One or more domains that share a common schema (which describes the information model in the directory) and a global catalog are referred to as a **forest.** If multiple domains in the forest have contiguous DNS domain names then that structure is referred to as a **domain tree.**

You create a domain by installing the first domain controller for a domain. During the installation of the first domain controller, the Active Directory Installation wizard uses information you provide to install the domain controller and create the domain within the existing context. This context may be the first domain in a new forest, the first domain in a new domain tree, or a child domain of an existing domain tree.

3.3 Domain Naming

Domains that form a single domain tree share a contiguous namespace (naming hierachy). Following DNS standards, the fully qualified domain name for a domain that is part of a contiguous namespace is the name of that domain, appended by the names of the parent and root domains, separated by the dot (.). For example, a domain with a NetBIOS name of “child” that has a parent domain named “torolab.ibm.com”, would have a fully qualified DNS domain name of “child.torolab.ibm.com”.

Domain trees in a forest do not share a contiguous DNS domain namespace.

3.4 Trust Relationship

Account authentication between domains is enabled by two-way, transitive trusts based on the Kerberos V5 security model.

Trust relationships is automatically created between adjacent domains (parent and child domains) when a domain is created in a domain tree. In a forest, a trust relationship is automatically created between the root domain of each domain tree added to the forest. Users and computers can be authenticated between any domains in the domain tree or forest.

It is possible to create a trust relationship between a Windows 2000 domain with pre-Windows 2000 domains by using **external trust.**

External trusts create trust relationships to domains outside the forest or to pre-Windows 2000 domains. External trust is one-way non-transitive trust. It is possible to create two-way trusts by creating two one-way trusts.

3.5 Kerberos V5 Authentication

Kerberos V5 is the primary security protocol for authentication within a Windows 2000 domain. The Kerberos V5 protocol verifies **both** the identity of the user and network services. This dual verification is known as **mutual authentication.**

The Kerberos V5 authentication mechanism issues *ticket* for accessing network services. These tickets contain encrypted data, including an encrypted password, that confirms the user’s identity to the requested service. The entire authentication process is invisible to the end user.

An important service within the Kerberos V5 is the **Key Distribution Center (KDC)**. The KDC runs on each domain controller as part of Active Directory, which stores all client passwords and other account information. A Kerberos client is installed on each Windows 2000 workstation and server.

4 DB2 UDB for Windows Authentication and Security

We have looked at Windows security and authentication. Now let's examine those concepts from a DB2 UDB for Windows perspective.

4.1 Authority Levels

An **authority level** is a set of rights to create or access database manager resources. These authorities can be assigned to a group of users.[2] The following figure illustrates the hierarchy of authority levels.

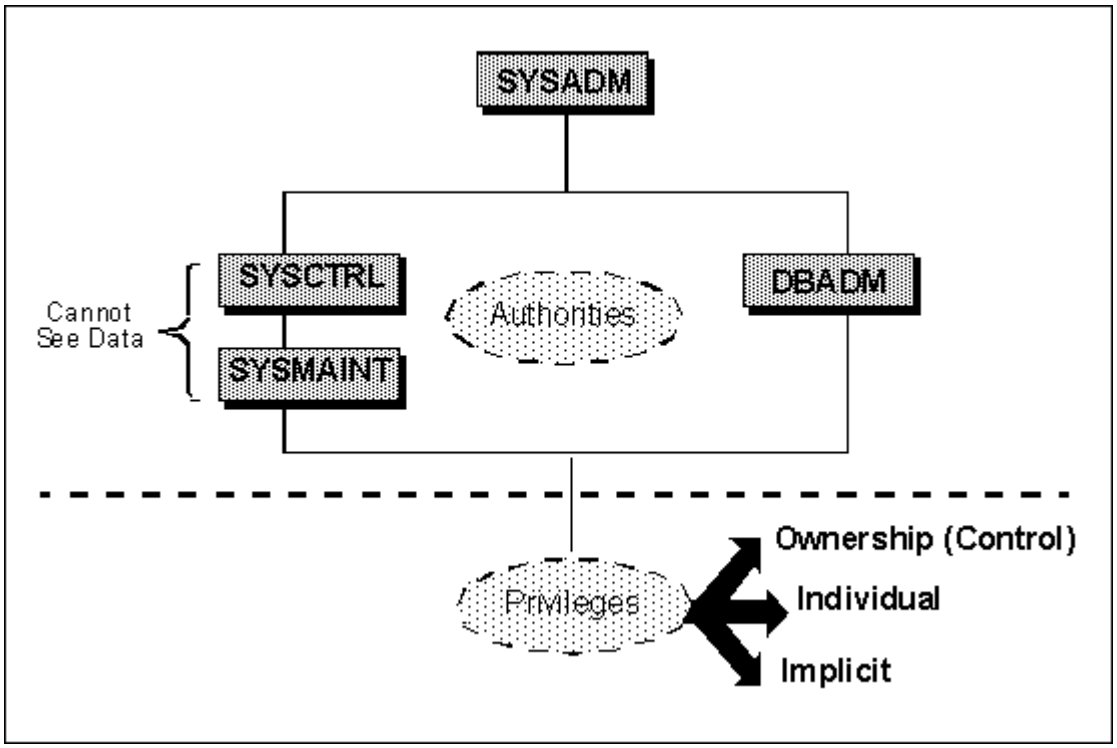


Figure 2. *DB2 Access Control Hierarchy*

At the top of this hierarchy is the DB2 System Administrator or **SYSADM**. Any user having SYSADM authority is able to perform any of the DB2 administration operations as well as access all database objects. Users with SYSADM authority are the only users allowed to configure the DB2 instance.

System administration authority for a particular DB2 instance is given to any user belonging to the NT group specified by the SYSADM_GROUP parameter in the DB2 database manager configuration file. When the instance is created, that parameter is set to null. Until the

[2] When a new instance of DB2 is created, none of these Authority Levels are associated with an NT group. You can define groups of users that you want to have a particular Authority Level, by creating the group in the Accounts Database and update the appropriate information in the Database Manager Configuration file.

SYSADM_GROUP parameter is given a value, user names that are members of the Administrators group[3] are considered to have system administrator (SYSADM) privileges. Generally you will not want all Windows NT Administrators to have SYSADM authority over your database instances, nor will you necessarily want your database administrators to be Windows NT administrators. Thus you will need to update the SYSADM_GROUP parameter promptly after installing DB2 UDB or after creating an instance. But before you do, ensure that the group exists. Use the Windows NT User Manager administrative tool to create groups.
Users that you wish to have SYSADM authority for the instance are members of that group.
To specify group names, update the database manager configuration file using one of the following methods:

• The Command Line Processor (or the Command Center), documented in the *DB2 UDB Command Reference.* For example:

```
db2 update dbm cfg using sysadm_group dbadmin
```

• The Control Center. Right click on the instance name, select *Configure...*, and select the *Administration* tab. Then enter the appropriate NT group name for *System administration authority group*.

• The configuration API, if you are developing your own application.

A user with SYSADM authority can perform both database manager and maintenance operations, as well as database operations. DB2 UDB provides two additional levels of system control authority, **SYSCTRL** and **SYSMAINT**. Like SYSADM, users receive these authority levels through membership in groups.

These group names must be entered in the database manager configuration file as values for the SYSCTRL_GROUP and SYSMAINT_GROUP parameters. Initially these parameters are set to null, meaning that no user has these authority levels.
SYSCTRL provides the ability to perform most administration commands. A member of the SYSCTRL user group does not have authority to access database objects nor modify the instance configuration file (DBM configuration). SYSCTRL offers almost complete control of database objects defined in a DB2 instance, but cannot access user data directly, unless explicitly granted the privilege to do so. A user with this authority, or higher, can perform the following functions:

- Update the database, node and DCS directory entries
- Update database configuration parameters
- Create or drop a database
- Force applications
- Start/Stop the DB2 instance
- Quiesce a database
- Execute the RESTORE/BACKUP/ROLLFORWARD commands
- Create or drop a table space

SYSMAINT authority allows the execution of maintenance activities but not access to user data. Only users with this level of authority (or higher) can do the following tasks:

- Update database configuration files
- Backup databases and table spaces
- Restore an existing database

[3] By default, SYSADM authority is held by the Administrator's group on the machine where the account is defined. For a Domain account this will be the Administrators group at the domain controller. As we will see later, this default can be changed in a couple ways.

- Restore table spaces
- Start and stop the DB2 instance[4]
- Run the Database Monitor
- Start and stop DB2 traces[5]

The authority levels of SYSCTRL and SYSMAINT provide access to instance-level commands and a limited number of database-level commands for the purpose of system maintenance. No direct access to data within the database is permitted for users that have only these authorities.

At the database administration level, there is the **DBADM** authority. The creator of a database will automatically have DBADM authority for the new database. Other users can be granted DBADM authority using the SQL GRANT statement. It is possible to hold DBADM authority for multiple databases. DBADM provides authority to perform some common administration tasks, such as loading data, creating database objects and monitoring database activity. A user with DBADM authority has access to data in that database as well.

A **privilege** is the right of a particular user or group to create or access a database resource. There are three types of privileges: Ownership, Individual, and Implicit.

1. **Ownership or control privileges**. For most objects, the user who creates the object has full access to that object. Control privilege is automatically granted to the creator of an object. There are some database objects, such as views, that are exceptions to this rule. Having control privilege is like having ownership of the object. You have the right to access the object and grant access to others. Privileges are controlled by users with ownership or administrative privileges. They provide other users with access using the SQL GRANT statement.
2. **Individual privileges**. These are privileges that allow you to perform a specific action. These privileges include select, delete and insert, and are granted by a user with ownership or control privileges.
3. **Implicit privileges**. An implicit privilege is one that is granted to a user automatically when that user is explicitly granted certain higher level privileges. They are not revoked when the higher level privileges are explicitly revoked.

We have talked about the possible classes of users and their related authority levels. Note that the first three (SYSADM, SYSCTRL, and SYSMAINT) are controlled outside of DB2 and recorded in the database manager configuration file, while DBADM is controlled within DB2 through the SQL GRANT statement and the SQL REVOKE statement.

The following table shows the valid authorities for the various DB2 levels.

Table 2. *Database Authorities*

Function	SYSADM	SYSCTRL	SYSMAIN	DBADM
CATALOG/UNCATALOG DATABASE	YES			
CATALOG/UNCATALOG NODE	YES			
CATALOG/UNCATALOG DCS	YES			
UPDATE DATABASE MANAGER CONFIGURATION FILE	YES			
GRANT/REVOKE DBADM	YES			
ESTABLISH/CHANGE SYSCTRL	YES			
ESTABLISH/CHANGE SYSMAINT	YES			
FORCE USERS	YES	YES		
CREATE/DROP DATABASE	YES	YES		
QUIESCE DATABASE	YES	YES		
CREATE/DROP/ALTER TABLESPACE	YES	YES		
RESTORE TO NEW DATABASE	YES	YES		

[4] Since starting and stopping an instance on NT implies starting and stopping the service that represents that instance, the user must also have the authority to start/stop a service.
[5] This is a general UDB statement, on Windows NT anyone can start trace.

UPDATE DATABASE CONFIGURATION FILE	YES	YES	YES	
BACK UP DATABASE/TABLESPACE	YES	YES	YES	
RESTORE TO EXISTING DATABASE	YES	YES	YES	
PERFORM ROLL FORWARD RECOVERY	YES	YES	YES	
START/STOP INSTANCE	YES	YES	YES	
RESTORE TABLESPACE	YES	YES	YES	
RUN TRACE[6]	YES	YES	YES	
OBTAIN MONITOR SNAPSHOTS	YES	YES	YES	
QUERY TABLESPACE STATE	YES	YES	YES	YES
UPDATE LOG HISTORY FILES	YES	YES	YES	YES
QUIESCE TABLESPACE	YES	YES	YES	YES
LOAD TABLES	YES			YES
SET/UNSET CHECK PENDING STATUS	YES	YES		YES
READ LOG FILES	YES	YES		YES
CREATE/ACTIVATE/DROP EVENT MONITORS	YES	YES		YES
USE IMPORT/EXPORT UTILITY	YES			YES

Assigning Authorities

SYSADM, SYSCTRL, SYSMAINT and DBADM should be restricted to avoid the risk of compromising system and data integrity. These authority levels are not required for general use. For a more detailed discussion of privileges and authorizations, see the *DB2 UDB Administration Guide.* For a complete list of the minimum authority level needed to execute DB2 commands and SQL statements, see the *DB2 UDB Command Reference* or the *DB2 UDB SQL Reference*.

4.2 Controlling Client Access to DB2 Databases

Controlling database access involves the control of access both to DB2 resources and to your data. A plan for controlling database access should be developed by defining your objectives for a database access control scheme, and specifying who shall have access to what resources and under what circumstances. Such a plan should also describe how to meet these objectives by using database functions, functions of other programs, and administrative procedures. For example, if a database contains sensitive data, database access must be planned carefully to allow access to items only when necessary. A plan for controlling database access must include the necessary actions to protect databases containing sensitive data and to ensure that the databases are physically secure.
This section describes how to control access to the database manager and how the database manager controls access within itself. It also describes how you can customize access to the databases in your instance. This section provides additional information about the needs and functions of different users and about using the system catalog tables to monitor access.

4.3 DB2 Authentication Methods

We've looked at how authentication occurs within Windows. DB2 UDB gives you some additional options as to where a user will be authenticated. You may wish user verification to occur at the client, at the server, or at a host. These options are provided through **authentication types**.

The authentication type is stored in the database manager configuration file at the database server. It is initially set to SERVER when the instance is created. All databases will have the same authentication type as the instance in which the database was created.

DB2 UDB provides the following authentication types:

[6] This is a general UDB statement on Windows NT anyone can start trace.

- **SERVER,**
- **SERVER_ENCRYPT,**
- **CLIENT,**
- **DCS,**
- **DCS_ENCRYPT,**
- **DCE, DCE_SERVER_ENCRYPT**
- **KERBEROS**
- **KRB_SERVER_ENCRYPT.**

Only the SERVER, SERVER_ENCRYPT, CLIENT, KERBEROS, and KRB_SERVER_ENCRYPT authentication types are discussed in this document.

• **SERVER.** Specifies that authentication occurs on the DB2 server. The user name and password must be specified during the connection or attachment attempt and are validated[7] at the server to determine if the user is permitted to access the instance. This applies to remote clients only, local applications can connect without providing a user name and password.

• **SERVER_ENCRYPT.** Specifies that authentication occurs on the DB2 server. The user name and password must be specified during the connection or attachment attempt and are validated at the server to determine if the user is permitted to access the instance. The password is encrypted before being sent to the server. This applies to remote clients only, local applications can connect without providing a user name and password.

• **CLIENT.** This authentication type provides **Single Signon.** Once a user signs on to the desktop no further authentication is required. A userid/password is not required on a connection or attachment request. If a userid/password is provided it is, by default, verfied on the client machine.

Since an explicit logon to the server is not required this authentication type is secure in an NT Domain where all machines are running Windows NT and all users log on using Domain accounts only. If all clients are NOT Windows NT clients you should consider using Server Authentication or limiting Client Authentication to **Trusted** Clients. Trusted Clients are defined to be clients that have a native security system. Specifically, all UNIX, OS/2, Windows NT, OS/400, OS/390, and VM clients have security systems and are regarded as **trusted**, whereas Macintosh, Windows 3.1, and Windows 95/98 clients are not. DB2 UDB has two database manager configuration parameters to help you deal with untrusted clients. These parameters are active ONLY when authentication for the remote instance is set to CLIENT.

1. *TRUST_ALLCLNTS*

   This parameter may be set to YES or NO. When set to YES, all clients are treated as trusted clients and a userid/password is NOT required on a connection or attachment request to a DB2 server. This is the default setting.

   If set to NO, untrusted clients must provide a userid and password when connecting to the server. Trusted clients will not have to provide a userid and password.

[7] It is important to understand DB2's validation method to avoid any confusion. When DB2 receives a userid it tries to find that account in the local SAM. If the account is not there and the machine is a member of a Domain, the Domain SAM will be searched. Failing to find it there all TRUSTED Domains will be searched. Once found, all group enumeration including the definition of an administrator will be accompished **on the machine where the account was defined.** As we will see later, the place group enumeration takes place can be changed.

If you are using Authentication CLIENT and you have untrusted clients connecting to your DB2 server, set TRUST_ALLCNTS to NO.

2. *TRUST_CLNTAUTH*

This parameter may be set to CLIENT or SERVER, and defaults to CLIENT. It determines where authentication will take place if a trusted client provides a userid and password on a connection or attachment request.

If set to CLIENT, the userid/password will be validated on the client machine. If set to SERVER, the userid and password are validated at the server.

Note that this parameter is invoked only if a userid and password are provided. If the client is trusted, and no userid or password are provided, authentication is assumed to have taken place at the client (single signon).

It is recommended that in a Domain Environment that if you decide on Client Authentication that you set TRUST_ALLCLNTS to NO and set TRUST_CLNTAUTH to SERVER. However if you have a mixture of NT and Windows 9x clients there are ways to set up the Windows 9x machines to force a DOMAIN logon.[8] In that case, it should be safe to specify TRUST_ALLCNTS as YES.

The major advantage provided by setting TRUST_CLNTAUTH to SERVER in a Domain is the significant reduction in RPC connections to the DOMAIN CONTROLLER if the client machines are connecting and providing a userid/password to be authenticated. The DB2 server machine will maintain one persistent connection to the Domain Controller.

• **KERBEROS.** Specifies to use Kerberos authentication. Kerberos authentication should only be used when both the client and server machines support Kerberos security protocol. This method is not supported for Windows NT. To use Kerberos authentication, both client and server machines must either belong to the same Windows 2000 domain or belong to trusted domains. The service account for the DB2 server service, if specified, must be a domain account. On the client machines, the user must logon to a domain account, or specify a domain user name during a connection or attachment. You will not be able to use local account for Kerberos authentication.

The Kerberos authentication type provides Single Signon. i.e. After the user logs on to the client machine using a domain account, any connection or attachment to a remote server does not require a user name and password to be specified.

Kerberos authentication works as follow:

1. User, when logging on to the client machine using a domain account, authenticates to the Kerberos Key Distribution Center (KDC) at the domain controller. The KDC issues a *Ticket-granting ticket* (TGT) to the client.
2. During the first phase of the connection, the server sends the target principal name, which is the service account name for the DB2 server service, to the client. Using the server's target principal name and the TGT, the client requests a *service ticket* from the *Ticket-granting*

[8] The Systems Policy Editor can be used to require a "Network Logon". Refer to the Windows 95 resource kit for more information. By defining the policy you can significantly reduce, but not eliminate the exposure provided by Windows 95.

service (TGS), which also resides at the domain controller. If both the client's TGT and the server's target principal name are valid, the TGS issues a service ticket to the client.
3. The client sends this *service ticket* to the server via the communication channel (e.g. TCP/IP)
4. The server validates the client's *service ticket* . If the client's service ticket is valid, then the authentication is completed.

It is possible to catalog the databases on the client machine and explicitly specify the Kerberos authentication type with the server's target principal name. This way the first phase of the connection can be bypassed.

If a user name and a password are specified, the client will request the TGT for that user account and use it for authentication.

• **KRB_SERVER_ENCRYPT.** Specifies that the server accepts KERBEROS authentication or encrypted SERVER authentication schemes. If the client authentication is KERBEROS and Kerberos authentication service is available, the client is authenticated using Kerberos authentication mechanism. If the client authentication is not KERBEROS or the Kerberos authentication service is not available, then the system authentication type is equivalent to SERVER_ENCRYPT.

The KRB_SERVER_ENCRYPT authentication type should be used when the server supports Kerberos (e.g. Windows 2000) and some, but not all, of the client machines support Kerberos authentication. This way, the client machines that support Kerberos will be authenticated using KERBEROS, and the ones that are not will be authenticated using SERVER_ENCRYPT.

4.4 DB2 UDB for Windows Group Support

DB2 UDB supports the use of groups in the granting of privileges and in the determination of Authority Levels (as discussed previously). DB2 will use the groups you define to NT, assuming they abide by the naming restrictions discussed below for user names.

DB2 UDB will allow you to specify either a Local group or a Global group when granting privileges or defining Authority levels. A user is determined to be a member of a group if the user's account is defined explicitly in the local or global group or implicitly by being a member of a global group defined to be a member of a local group.

4.4.1 Group Enumeration

DB2 UDB will determine whether a user account is defined in the local machine's SAM, the SAM of the Domain Controller or the SAM of a Domain Controller in a Domain trusted by the Domain containing the DB2 UDB Server.[9]

By default, DB2 will then go to a machine where that account is defined to enumerate groups. If that account is a Domain account then the machine
is a Domain controller for that Domain. This means that the Domain Administrator is responsible for defining groups that are to be used by DB2.

DB2 UDB can be configured to enumerate groups on the local machine rather than on the Domain Controller. This creates an extra level of administration overhead but allows the DB2

[9] As of DB2 UDB Version 7.1 if the server is configured with Client Authentication and the Client is a Windows Client that is part of a Domain, DB2 will flow the Domain Name as well as the User Name to the server. In this case the standard search order of Local Sam, Domain Sam and Trusted Domain Sam will be bypassed.

Administrator (assuming he is also a local Administrator), to control the groups that DB2 UDB sees. To have DB2 look up groups locally use the DB2SET command to define DB2_GRP_LOOKUP.

On the DB2 Server machine

To set it globally (for all instances of DB2 Server):

```
db2set -g db2_grp_lookup=local
```

To set it for the current instance on a DB2 server:

```
db2set –i instancename db2_grp_lookup=local
```

After issuing this command, you must stop and start the DB2 instance(s) for the change to take effect.

Then create local groups and include domain accounts (domain\accountname) or global groups in the local groups.

4.4.2 DB2_GRP_LOOKUP=DOMAIN

If you have selected a Master Account Domain Model (with DB2 in a Resource Domain) and decided to use the Domain controller as the place where DB2 will enumerate groups then DB2 must determine the name of the machine that is the Domain Controller for that other Domain. In and NT 4.0 environment, DB2 will always find the Primary Domain Controller. However if the PDC is unavailable DB2 may or may not be able to locate a suitable Backup Domain Controller. If DB2 is running on a machine that is a Primary or Backup Domain Controller in the Resource Domain, it will be able to locate any Domain Controller in any Trusted domain. This is because the names of Domains of Backup Domain Controller in Trusted Domains are only known if you are a DOMAIN controller.

If DB2 is not running on a Domain Controller then you should probably issue:

```
db2set -g db2_grp_lookup=domain
```

This will tell DB2 to use a Domain Controller in its own Domain to find the name of a Domain Controller in the Accounts Domain. In other words, when DB2 finds out that a particular user account is defined in DOMAINX, rather than trying to locate a domain controller for DOMAINX, it will send that request to a Domain Controller in its own Domain. The name of a Domain Controller in the Account Domain will be found and returned to the machine that DB2 is running on. This has two advantages:

1. **It will find a Backup Domain Controller when the PDC is unavailable.**
2. **It will find a Backup Domain Controller that is close when the PDC is geographically remote**

4.5 The DB2 for Windows Environment

This section discusses some of the considerations that you need for logging into a DB2 for Windows environment, especially the first time the database manager is started. You'll need to understand the restrictions that DB2 imposes on user and group IDs and passwords. Then, you'll need to understand the default Windows environment and what to change before logging into DB2.

4.5.1 User ID and Group ID Limitations

Once a user is authenticated, the user is identified within DB2 using an SQL authorization name **authid**. The authid is used to track authorities and privileges and is used as the default high-level qualifier for database object identification. On the Windows NT platform, DB2 UDB uses the Windows NT userid as the DB2 authid. Therefore, userids that will be used to connect to DB2 databases must follow certain DB2 naming conventions.

For Version 7.1 or later, the user names are limited to a maximum of thirty characters. Earlier versions of DB2 only support user names of eight characters or less. Group ids are limited to a maximum of eight characters. Passwords are limited to a maximum of fourteen characters. There are other restrictions as well:

- Cannot start with a digit (0 to 9) or end with a dollar sign ($).
- Can be one to eight characters long and may contain the following characters:
  - Upper or lower case characters A to Z.
  - Special characters #, @ or $.
  - Digits 0 to 9.
- Cannot be PUBLIC, USERS, ADMINS, LOCAL or GUESTS, or any SQL reserved word, or a name that starts with IBM, SYS or SQL.

Note:
Passwords in Windows NT are case-sensitive, although user IDs are not. This can be a common cause for logon failure.
If you use domain accounts then ensure that the local machine name running the DB2 Server is different then any domain account. This is required because of the Windows search order. It will return a local account or the local machine name if a match is found, prior to searching a Domain Controller.

4.5.2 Authority to Install DB2 UDB for Windows NT

To install DB2 UDB or start it the first time, you need to log on to Windows NT with an administrator user ID. The Windows NT default administrator (Administrator), although longer than eight characters, is now supported for installation. You must create another account that conforms to the DB2 user and group ID limitations, and is a member of the administrator group on the machine in which DB2 for Windows NT will be installed. This account can then be used as the DB2 Systems Administrator (SYSASM).

4.5.3 Authority to start DB2 UDB for Windows

The DB2START command launches DB2 as a Windows Service. It also is run as a service when started from the Windows Control Panel / Services dialog box or with the NET START command.

Because DB2START launches a Windows service, a user attempting to start DB2 must meet Windows NT's requirements for starting a service by being a member of the Administrators, Server Operators or Power Users group.

4.5.4 Service account requirements

During the installation of DB2 on Windows NT or Windows 2000, the setup program creates several Windows services and assigns a service account for each service.

The following DB2 services have service accounts:
- DB2 – DB2
- DB2 – DB2CTLSV
- DB2 – DB2DAS00
- DB2 Governor
- DB2 JDBC Applet Server – Control Center

In order to run DB2 properly, the setup program grants the following User Rights (assignable through the User Manager Policies/User Rights dialog) for the service account that is associated with the DB2 server services:

1. Acts as part of the operating system
2. Create a token object
3. Increase quotas
4. Log on as a service
5. Replace a process level token
6. Lock pages in memory[10]

These rights are required so that the DB2 service can authenticate users and enumerate users and groups from a domain controller. If you wish to use a different service account for the DB2 service(s), you must grant the above User Rights to the service account.

In addition to the above User Rights, the service account must also have write access to the directory where the DB2 product is installed.

The service account for the DB2 Administration services (i.e. DB2DAS00 service, DB2 Governor, and DB2 JDBC Applet Server – Control Center) must also have the authority to:
- Start and stop other DB2 services (i.e. the service account needs to belong to the Power Users group)
- Have DB2 SYSADM authority against any DB2 instances that it administers.

4.5.5 DB2 for Windows NT Security Server Service

The DB2 for Windows NT Security Service is installed as a part of the Client Application Enabler for Windows NT. The Security Service is required only on Windows NT client machines when connecting those clients to a server that is configured for authentication CLIENT and TRUST_CLNTAUTH set to CLIENT and you specify a userid/password on the connect.[11]

The service is normally configured to autostart.
The service can be manually started (and stopped) in two ways.

1. The first is to enter the following command from a command window:
   NET START DB2NTSECSERVER

[10] The *Lock pages in memory* user right is only needed for Address Windowing Extension (AWE) support. DB2 uses AWE API to support large memory under Windows 2000 Advanced Server and Windows 2000 Data Center Server.
[11] This also applies to applications run on a DB2 server machine that are connecting to a server on another machine.

2. Open Services within the Control Panel. Select **DB2 Security Server** by clicking on it and then select the **Start (Stop)** button

To have the service start automatically with Windows NT, open Services within the Control Panel. Select DB2 Security Server and then select Startup. A Service window will appear. Click on the Automatic radio button under Startup type and click on **OK**. Note that you have to be logged onto the machine with an account that is a member of the Administrators group to change the Services configuration.

5 Planning for DB2 UDB in a Windows Environment

This section looks at where you might install DB2 UDB Server in a Windows NT environment and looks in detail at what is required if it is installed on a Backup Domain Controller.

An important point when considering where to install DB2 is the type of authentication that will occur between clients and server. This will partly be driven by the types of clients on the network (see “DB2 Authentication Methods” for notes on trusted clients). Also consider the types of clients in any trusted domains.

5.1 DB2 UDB for Windows NT in a Workgroup

If you are considering installing DB2 for Windows NT in a workgroup first consider if a workgroup is suitable for your needs. DB2’s notion of Client level security provides Single Signon but in a Workgroup there are many SAM databases and there are no guarantees that when you grant a privilege to LESLIE that a LESLIE account will not exist on more than one client machine. But despite this exposure, Client level security is convenient[12] for many small offices where access to data is either not restricted or can be controlled by the types of programs that the users can access. If you need tighter security then the only option is to set the DB2 Instance to Server level Security. This requires that the user provide a userid/password during Connect and you maintain duplicate accounts on each DB2 Server. In either case, Groups are enumerated on the DB2 Server machine.

5.2 DB2 UDB for Windows NT in a Domain

There are two basic options for installing DB2 UDB for Windows NT in a domain. They are:

3. **On a workstation. This could be a Windows NT Server that is not a domain controller or a Windows NT Workstation where Peer-to-Peer services are enabled.**

4. **On a Domain Controller. In a Windows NT LAN environment, a user can be authenticated at either a Primary or Backup Domain Controller.**

This ability to authenticate on multiple machines is very important in large distributed LANs with one central PDC and one or more BDCs at each site.

5.2.1 DB2 UDB for Windows NT on a Non-Domain Controller

One of the big advantages of putting DB2 UDB for Windows NT on a dedicated server is that it can be tuned and secured specifically for that task. Also the DB2 application would not have to compete for server resources with domain controller activities.

On the negative side, regardless of which type of authentication DB2 is configured for, Windows NT authentication may have to occur on another server in its own or another domain with the delays associated with communicating with another machine while processing a connnect request.

[12] Single Signon is provided, and if you choose not to use groups you can eliminate the need for duplicate account maintenance.

5.2.2 DB2 UDB for Windows NT on a Primary Domain Controller

DB2 could be installed on a Primary Domain Controller. The biggest advantage of installing DB2 UDB for Windows NT on a PDC is that authentication can occur on the same machine. Also, there is less chance of encountering difficulties with the resolution of group membership for user accounts.

Implementing DB2 on a PDC also avoids some of the extra configuration that is discussed in the next section. However, it is recommended, especially in larger networks, that the Primary Domain Controller be maintained as a dedicated server.

Once installed and configured, DB2 administrators and Windows NT administrators can and should be kept separate. In a larger enterprise or network, it would not be desirable, from a domain administration point of view, to have DB2 administrators logging on and performing DB2 maintenance on the PDC. The opposite is also true from a DB2 viewpoint. You would not want PDC administrators logging into a DB2 instance as SYSADM.

5.2.3 DB2 UDB for Windows NT on a Backup Domain Controller

Installing DB2 for Windows NT on a Backup Domain Controller offers all of the same advantages as those for a Primary Domain Controller. Users can be authenticated on the BDC at their site (and, in fact, on the same machine) in a distributed environment instead of requiring a call to the PDC for authentication. However, in order to enable DB2 to use the security database on the BDC, you must set a DB2 profile variable called DB2DMNBCKCTLR.

If the DB2DMNBCKCTLR profile variable is not set or is set to blank, DB2 UDB for Windows NT performs authentication at the PDC.

Set DB2DMNBCKCTLR to the name of the domain that the BDC is a member of. For example,

DB2SET DB2DMNBCKCTLR=DB2NTDMN

If the DB2DMNBCKCTLR profile variable is set to a question mark
(that is, DB2DMNBCKCTLR=?), then DB2 UDB for Windows NT will attempt to determine the domain that this machine is a BDC for. If you know the domain name, then just use the first method.

Usage Scenarios

In the following scenarios we will use the following machines:

- WGCN– Workgroup NT Client Machine
- WGCW – Workgroup Windows 9x client machine
- WGDB – Workgroup DB2 Server Machine
- PDC – Primary Domain Controller
- BDC – Backup Domain Controller
- TPDC – Trusted Domain Controller
- DMDB – Domain DB2 Server Machine
- DMCN – Domain NT Client Machines

We also assume that the Connect Priviledge has been granted to PUBLIC, which is the default when a database is created.

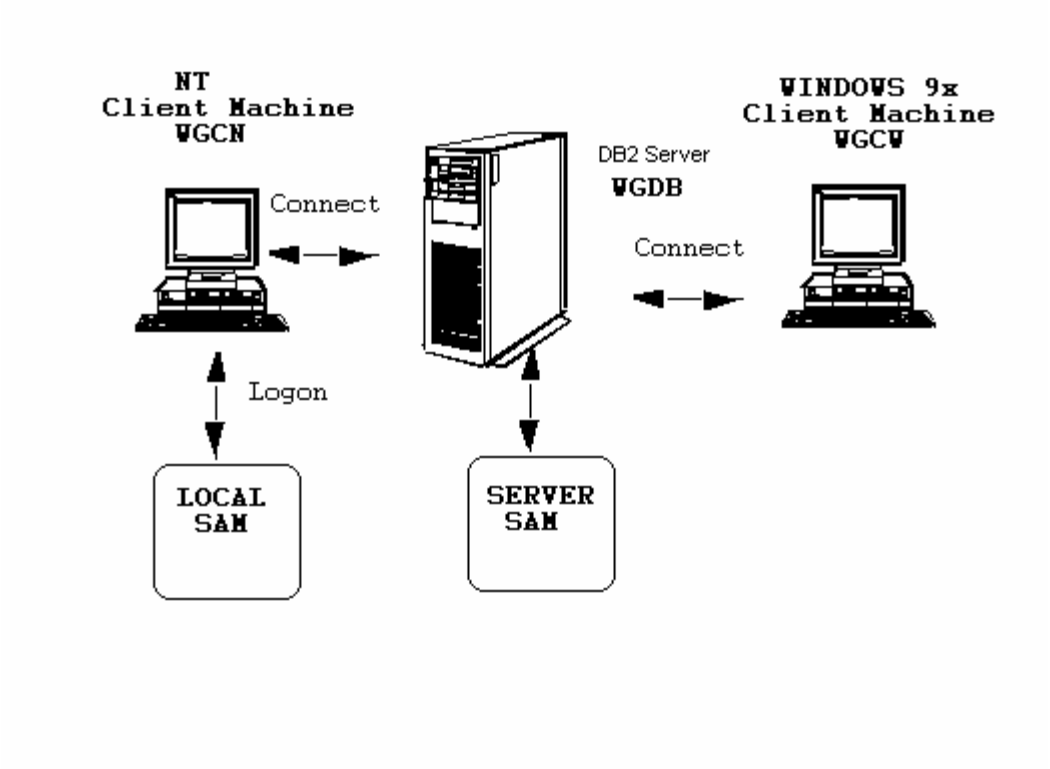
5.3 Client Authentication in a Workgroup Environment

- When user Dale on WGCN connects to WGDB and does not provide a userid/password DB2 extracts the userid from the operating system and sends it to the server. WGDB searches the local SAM to determine if Dale is known. If the account is NOT found then groups are NOT enumerated and Dale has access to all data granted to PUBLIC or to DALE explicitly.

- When user Dale on WGCN connects to WGDB as user FRED with password FREDPW, DB2 will attempt to validate the password on the Client Machine. The DB2 Security Service will be contacted to do the validation. A common cause of error in this scenario happens when the DB2 security service is not running, the application **has SQL1402 Unable to authenticate user due to unexpected system error**, returned. If the account is found and the password is valid the userid FRED is sent to WGDB and processing continues as discussed above.

- When user Dale on WGCW connects to WGDB as user FRED with password FREDPW, DB2 cannot validate the password since Windows 9x does not have a native security system. DB2 trusts that the password provided is valid and passes the userid FRED to WGDB and processing continues as discussed above.

In the scenarios above, if you want to use local groups then you MUST create duplicate user accounts on WGDB. If you want to prevent the user on WGCW from accessing WGDB without authentication, set TRUST_ALLCNTS to NO. If you don't want to run the security service on the client machine WGCN and want the userid/password validated at the server set TRUST_CLNTAUTH to SERVER.



5.4 Server Authentication in a Workgroup Environment

- When user Dale on WGCN connects to WGDB and does not provide a userid/password DB2 extracts the userid from the operating system and sends it to the server. WGDB determines that the DB2 instance has server authentication and rejects the request. **Note that if the application were running on the WGDB machine that a userid/password would NOT be required even though the authentication type is SERVER.**

- When user Dale on WGCN (or on WGCW) connects to WGDB as user FRED with password FREDPW, DB2 will validate the password on the Server Machine. The account MUST exist on the WBDB or the connection request will fail. Once the account is validated, groups and authorities will be enumerated by using the Security Database on WGDB.

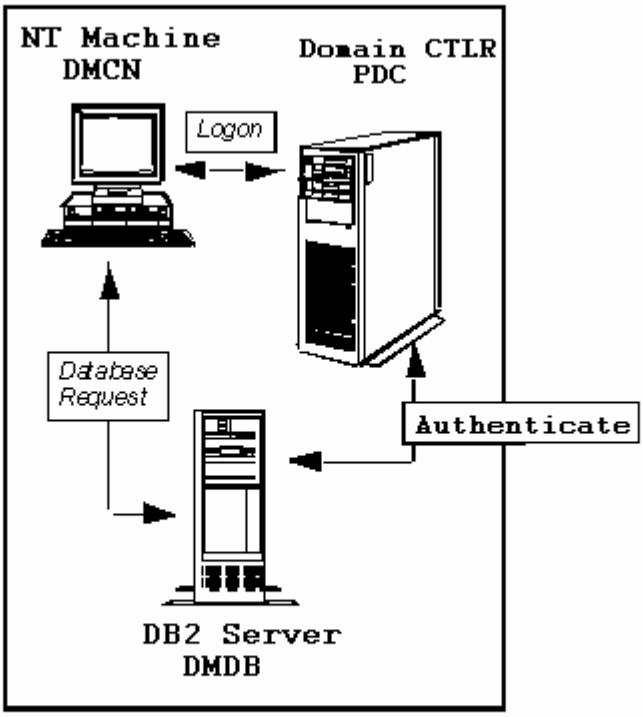
5.5 Client Authentication in a Single Domain Environment

- When domain account Dale on DMCN connects to DMDB without providing a userid/password, the account name is passed to DMDB. DB2 will search the local SAM and if the account is not found it will search the Domain Controller. When the account is known to be a domain account, DB2 will communicate with the PDC machine to enumerate groups. The search order described above is another common source of error. DB2 does not flow domainname/userid to the server, only the userid is sent. If the userid exists in both the local SAM and in the Domain SAM. DB2 will find the local userid first and enumerate groups on the local machine. Therefore, ensure that **there are no accounts on the local server that have the same name as accounts in the DOMAIN.**

- When domain account Dale on DMCN connects to DMDB as FRED using password FREDPW the account name and password is verified on the client machine by communicating with the Security Service. The Security Service will search the local SAM and if the account is not found it will search the Domain Controller. When the account is

known to be a domain account, DB2 will communicate with the PDC machine to validate the password. If the password is valid, DB2 will pass the account FRED to the Server machine and processing will continue as described above. If TRUST_CLNTAUTH was set to SERVER then the userid/password would be validated at the Server machine which is most likely the desired behavior since it would dramatically reduce the number of RPC connections required to the PDC machine.

If the DBA of the DMDB machine wished to control the definition of groups and potentially eliminate load on the PDC, he could create local groups on the DMDB machine and include NTDOMAIN\Dale and NTDOMAIN\Fred and set DB2_GRP_LOOKUP to local. Many installations choose this option since it allows the separation of DB2 groups from system groups defined by the Domain Administrator and does not preclude the inclusion of Domain GLOBAL GROUPS in the local groups on DMDB.



5.6 Server Authentication in a Single Domain Environment

- When domain account Dale on DMCN connects to DMDB without providing a userid/password, the account name is passed to DMDB. DMDB determines that the DB2 instance has server authentication and rejects the request. **Note that if the application were running on the WGDB machine that a userid/password would NOT be required even though the authentication type is SERVER.**

- When domain account Dale on DMCN connects to DMDB as Dale using DALEPW the account name and password is verified on DMDB. DB2 will search the local SAM and if the account is not found it will search the Domain Controller. When the account is known to be a domain account, DB2 will communicate with the PDC machine to validate the password. If the password is valid, DB2 will continue to enumerate groups on the Domain Controller.

As discussed with Client Authentication, the DB2 administrator can change where groups are enumerated by setting DB2_GRP_LOOKUP to local.

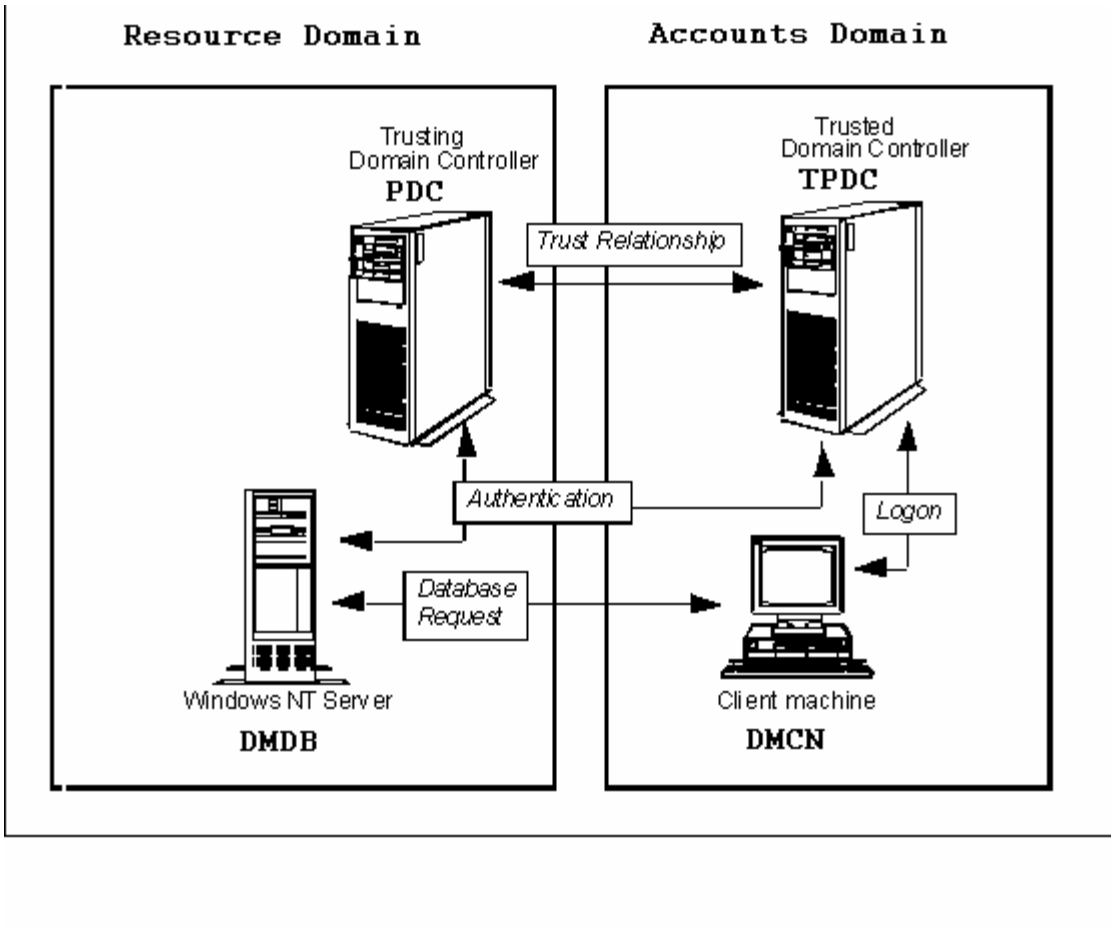
5.7 Client Authentication in a Master Account Domain Environment

The examples discussed under the Single Domain Environment apply to the MAD environment, except that DB2 will enumerate groups across the trust relationship. The trust relationship does not have to be two-way. All that is required is that the Resource Domain trusts the Account Domain and the Account Domain is setup to allow itself to be Trusted by the Resource Domain.

As an illustration, I will discuss the default Client security scheme as it applies to a Master Account Domain.

User Dale logs on to a domain account in the Master Account Domain. The machine DMCN is in this domain, the Domain controller for this Domain is TPDC. The DB2 server DMDB is in the resource domain whose Domain Controller is PDC. A trust relationship exists between TPDC and PDC. When Dale connects to a database on DMDB, the userid is sent to the DB2 server. DB2 searches the local SAM, then the SAM of the PDC in the resource Domain and finally the SAM in the Master Account Domain. The account is found and the machine TPDC is used to enumerate groups. Again a common problem in this environment is to have duplicate accounts in either the Local Sam or in the Resource Domain SAM. For this to work properly, **ENSURE a single name space.**

As described previously the DBA on DMDB can decide to control group membership (and reduce connection overhead) by set DB2_GRP_LOOKUP to local.



6 Frequently Asked Questions

- **All my accounts are domain accounts but I do not want the Domain Administrator controlling the definition and membership of groups that my DBA will use in the granting of DB2 privileges.**

  What you should do is create local groups on the DB2 Server and in these local groups explicitly specify the domain accounts that you want to include (ie domainname/account). You can include global groups if you choose but then you are affected by any changes made to those global groups by the Domain Administrator. Once you create your local groups set the DB2 profile variable DB2_GRP_LOOKUP=local and restart the DB2 Instance. All group enumeration will be performed on the local machine (this includes the definition of DB2 authority levels).

- **How do I get someone other than a local administrator or a Domain Administrator to be an administrator of DB2.**

  You should create another group, for example DB2ADM, and include in it the accounts you want to administer DB2. Then update the Database Manager

Configuration file to set sysadm_group to DB2ADM. The same principle applies toe the syscntrl or sysmaint definitions.

- **I log on to a Domain Account but DB2 can't find the groups I am a member of.**

  Remember that DB2 does not qualify the name of a user with the domain name. So when the account is presented to the server, DB2 searches the local SAM, followed by the Domain SAM followed by the SAM of trusted Domains. Once the account is found groups are enumerated on that machine. A common source of problem is to have a local account the same as a Domain account. DB2 finds the account on the Local machine and searches the local SAM for groups. Delete the local account.

- **My Domain Account is a member of the local Administrator's group but I can't Administer DB2.**

  There are two possibilities. The most likely is that since DB2 has found your account on the Domain Controller it is going to the Domain Controller to determine if you are an Administrator. Since you are NOT a domain administrator your are not, be default, a DB2 Administrator. To get around this you can tell DB2 to look on the local machine for its group definition (DB2_GRP_LOOKUP=local) or you can define an alternate group to be used as the SYS_ADM group and update the DBM Configuration to reflect this group.

  The other possibility is that you are not a member of the group defined in the DBM Configuration file ad SYS_ADM (this overrides the use of the Administrators group).

- **I define all my domain accounts in local groups but DB2 doesn't seem to recognize them.**

  You should set DB2_GRP_LOOKUP=local to have DB2 look on the local machine for group definitions. By default, it looks on the machine where it finds the definition of the account (the Domain Controller).

- **Why can't DB2 execute my stored procedure or User Defined Function (UDF)?**

  The most common cause of this problem is that the stored procedure or UDF reside on a LAN DRIVE or there is a LAN DRIVE in the path and the search order is such that the stored procedure or UDF itself or one of its dependencies are in a directory that is specified after that LAN DRIVE. This problem usually occurs because the DB2 Service, by default, runs under the system account. The System Account cannot access LAN drives. The way to correct this is to change the DB2 Service to run under a specific administrator ID that has the advanced user rights discussed earlier in this document.