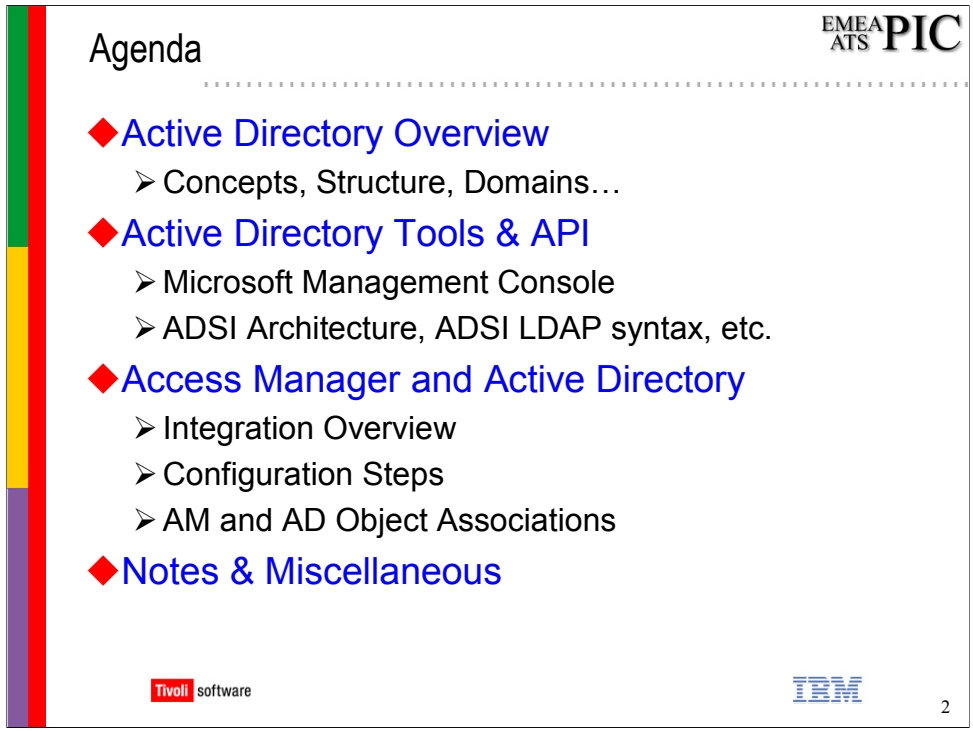
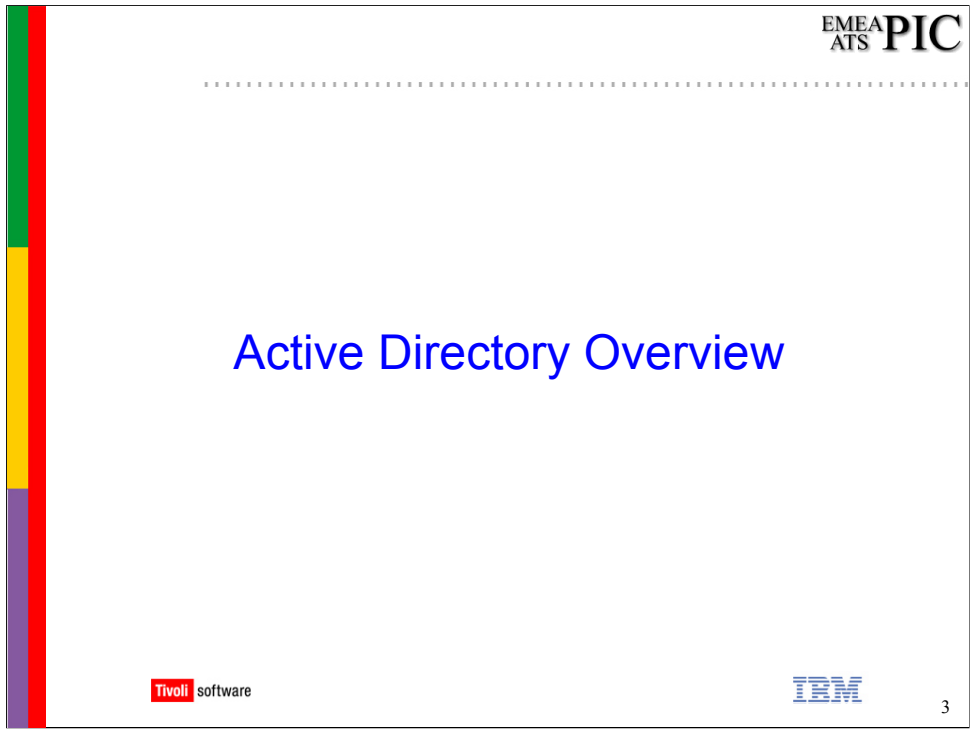
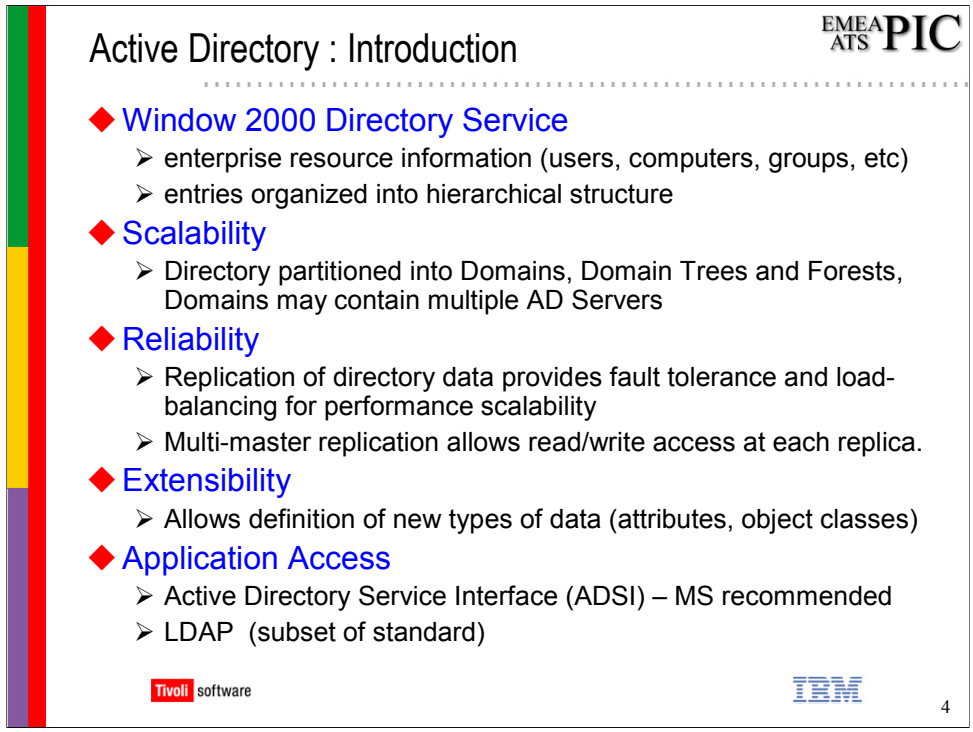

Please be aware that additional information may be found in the speaker notes of this presentation.




This presentation covers the concepts of Active Directory, the supported programming interfaces and the configuration of Access Manager 3.9 to use a single-domain or multiple-domain Active Directory.

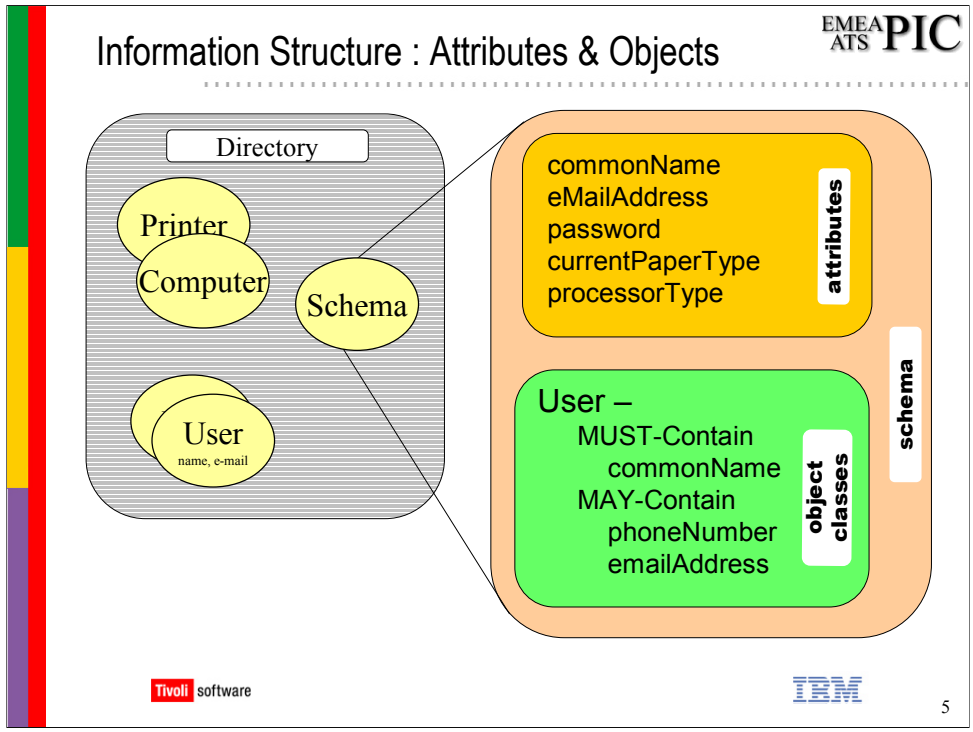

EMEA ATS PIC
Active Directory Overview
Tivoli software
IBM
3




A *domain controller* is a Windows 2000 Server that is configured to run the Active Directory service.

The LDAP supported by AD is a subset of LDAP because of the lack of mechanism to control the ACLs of the object.

Furthermore, access to Active Directory objects is controlled by a Security Descriptor (SD) that is associated with each object. The SD specifies which user/group is allowed to perform operations to the object. The SD is stored under the *nTSecurityDescriptor* attribute in each object in binary format. Because the SD is binary, the only way to read/write/modify the SD is via a call to an interface defined by Microsoft, which understands the format of the SD value.




LDAP and AD have many similar concepts. Both organize the directory information into *objects* or *entries* (AD tends to use the term "object", LDAP tends to use the term "entry").

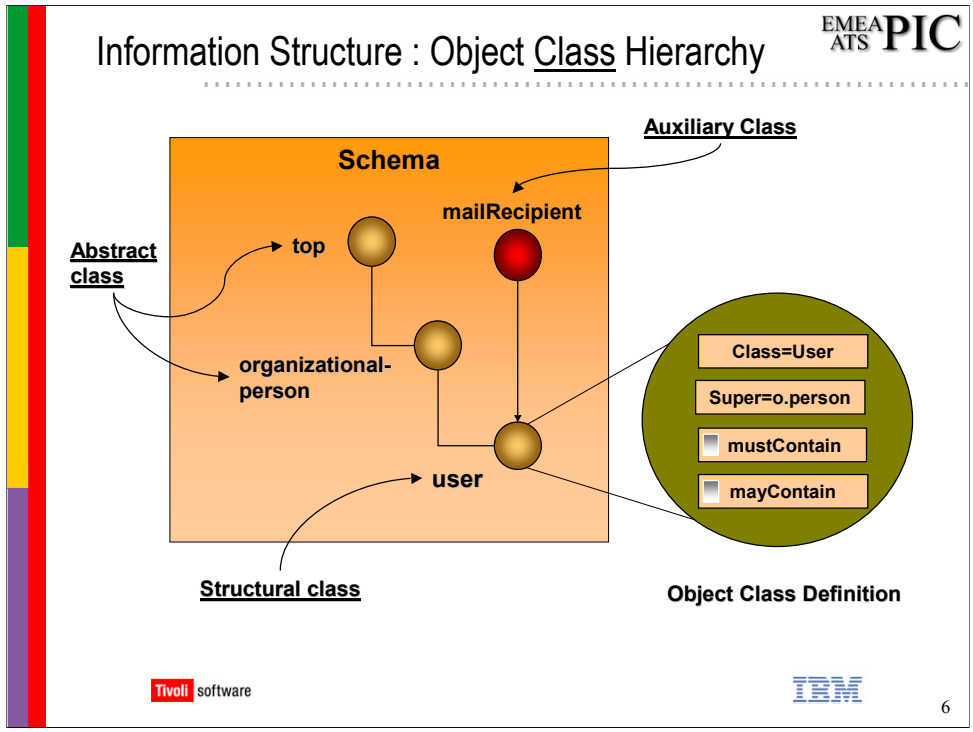
The most basic information structure is called an *attribute*. E.g., 'phone-number', 'Address, 'e-mail' etc.

- Attributes can have a single value, like emailAddress: "me@company.com", or
- Attributes can have multiple values like member-of: "groupA", "groupB", "groupC".
- The binary structure of the attribute is called its *syntax*.

A collection of attributes is called an *object class*. E.g., 'User', 'Organization', 'Printer', etc. The definition of an object class specifies some attributes that **must** be present and others that **may** be present. **It is objects that are stored in the directory**.

The directory *schema* refers to all the Attribute and Object Class definitions together. The schema defines what types of information can be stored, how the information is organized, and the rules for when an object is complete.

- Sometimes the schema is referred to as *metadata* – "data about the data".
- The schema data is stored in the AD. Updates to the schema are made to the *schema operations master* (one and only one in AD Forest) which then replicates the changes to other AD Servers.



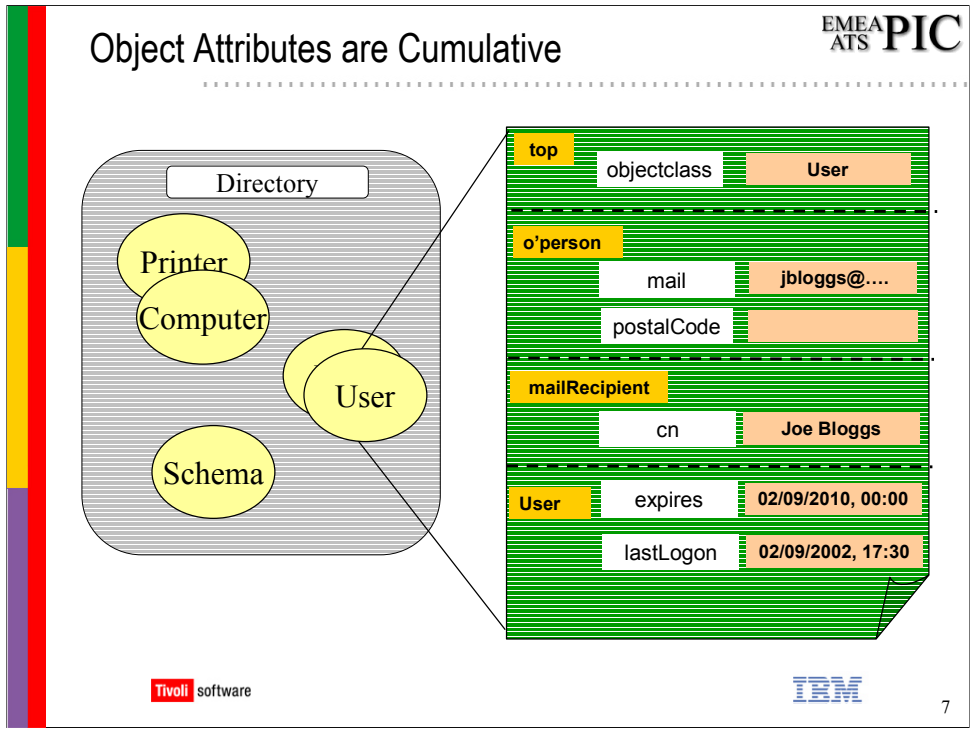

Object classes have a hierarchy that is created by *deriving* one object class from another. For example, the *Printer* object class might be derived the *Peripheral* object class. The new class will have its own attribute definitions plus those that it "inherits" from its *parent* or *superior* class. That is, a 'printer' object will have all of the attributes defined for 'peripheral' class plus whatever attributes are defined on the printer 'class'.

Each object classes is defined as one of these types:

- *structural* - xxxx
- *abstract* - xxxxx
- *auxiliary* - are just attribute collections – cannot be created in directory.

All object classes derive (directly or indirectly) from a special class called "top"

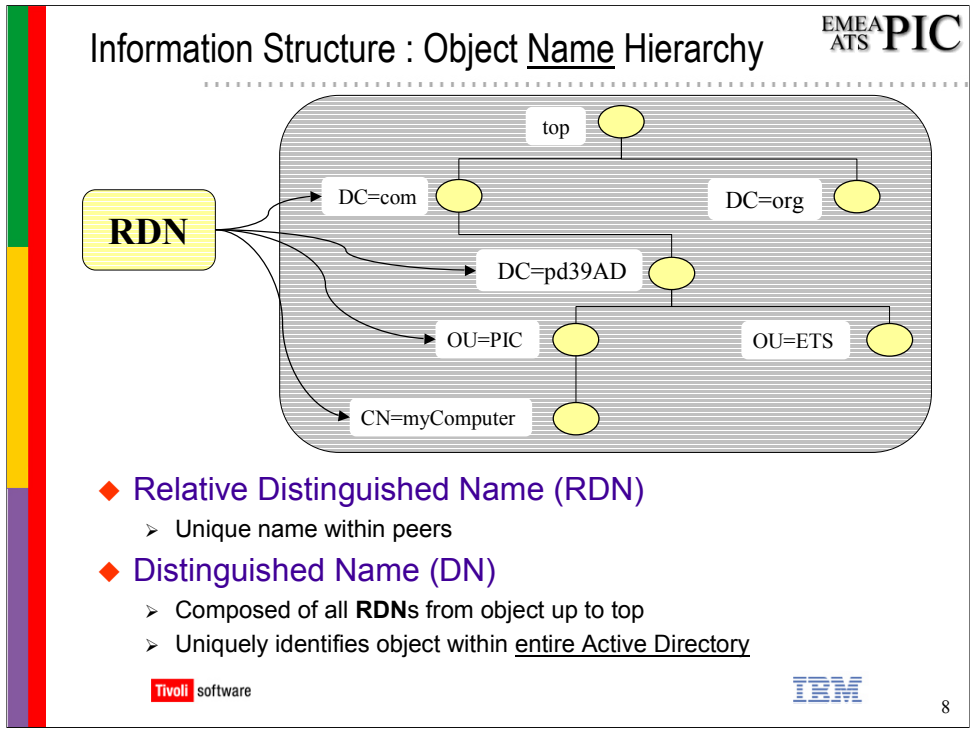
To browse the schema definition in AD you must have installed the 'Schema snap-in' on Win2K Server.




This diagram shows that an object within the registry contains attributes from all object classes that contribute to the definition of its object class.

E.g, because the Object Class 'User' is derived from 'organizationalPerson' which is derived from 'top', a User within the registry contains attributes from all those classes. Because object class 'User' is also derived from the auxiliary class 'mailRecipient', a User object contains attributes from that class too.

So you can see how an object class can be considered a *template* for objects of that type.




Of course, having objects in AD is only useful if you can find them! That is why every object is required to have a *Distinguished Name.*

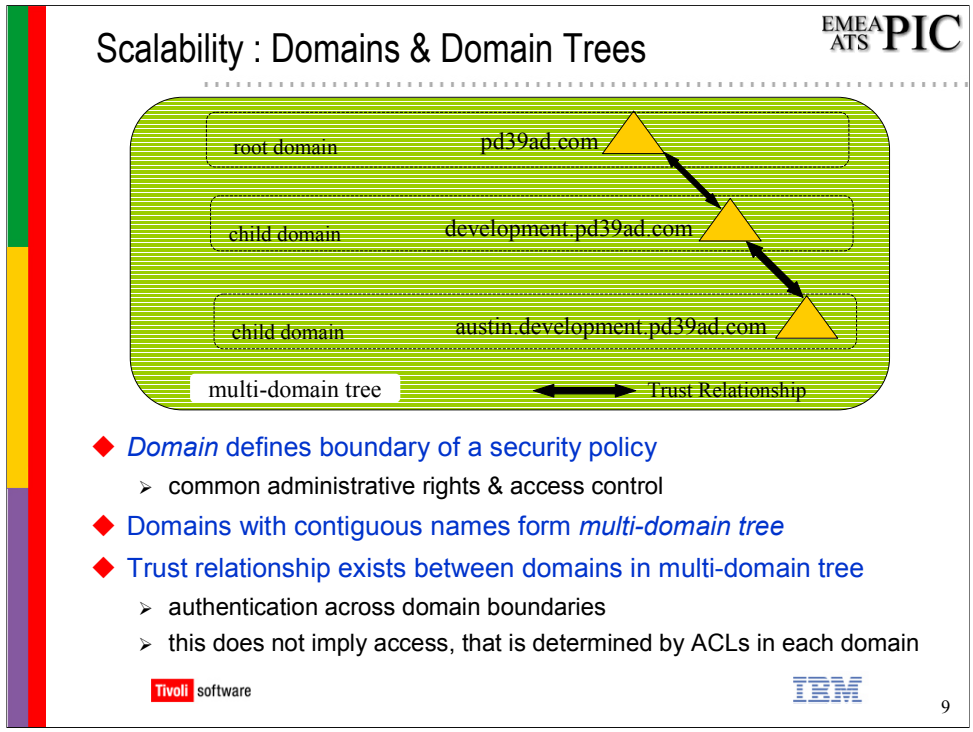
Each object in Active Directory can be referenced by several different names. AD creates a relative distinguished name and a canonical name for each object using information provided when the object was created or modified. Each object can also be referenced by its *distinguished name*, which is derived from the relative distinguished name of the object plus the distinguished name of its parent object.

•The LDAP *relative distinguished name* (**RDN**) uniquely identifies the object within its parent container. For example, the RDN of a computer named "myComputer" is CN=mycomputer.

•The LDAP *distinguished name* (DN) is globally unique. For example, the distinguished name of a computer named 'myComputer' in the 'PIC' organizational unit in the 'pd39ad.com.com' domain is "CN=myComputer, OU=PIC,DC=pd39AD,DC=com"

•The *canonical name* is constructed the same way as the distinguished name, but it is represented using a different notation. The canonical name of the computer in the previous example would be "pd39ad.com/PIC/myComputer".

Access Manager always uses the LDAP form of object name.




A domain defines a boundary of security policy. The AD includes one or more domains, each having its own security policies and trust relationships with other domains.

A Domain defines security policies and settings - such as administrative rights and access control lists. These do not cross from one domain to another.

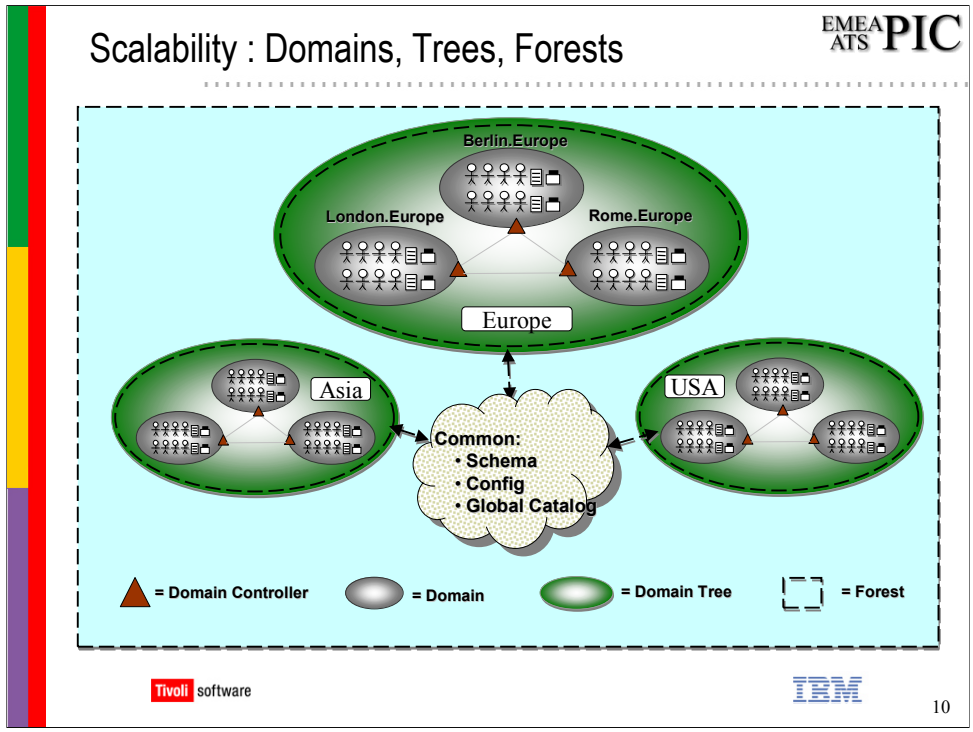
Each domain stores only the information about the objects within that domain. By partitioning the directory this way, Active Directory can scale to very large numbers of objects.

A domain defines a scope or unit of policy. A *Group Policy* object establishes how domain resources can be accessed, configured, and used. These policies are applied only within the domain and not across domains.

Domains are units of replication. All of the domain controllers in a particular domain can receive changes and replicate those changes to all other domain controllers in the domain.

Some reasons to create more than one domain are:

- Different password requirements between departments or divisions
- Massive numbers of objects
- Different Internet domain names
- More control of replication
- Decentralized network administration

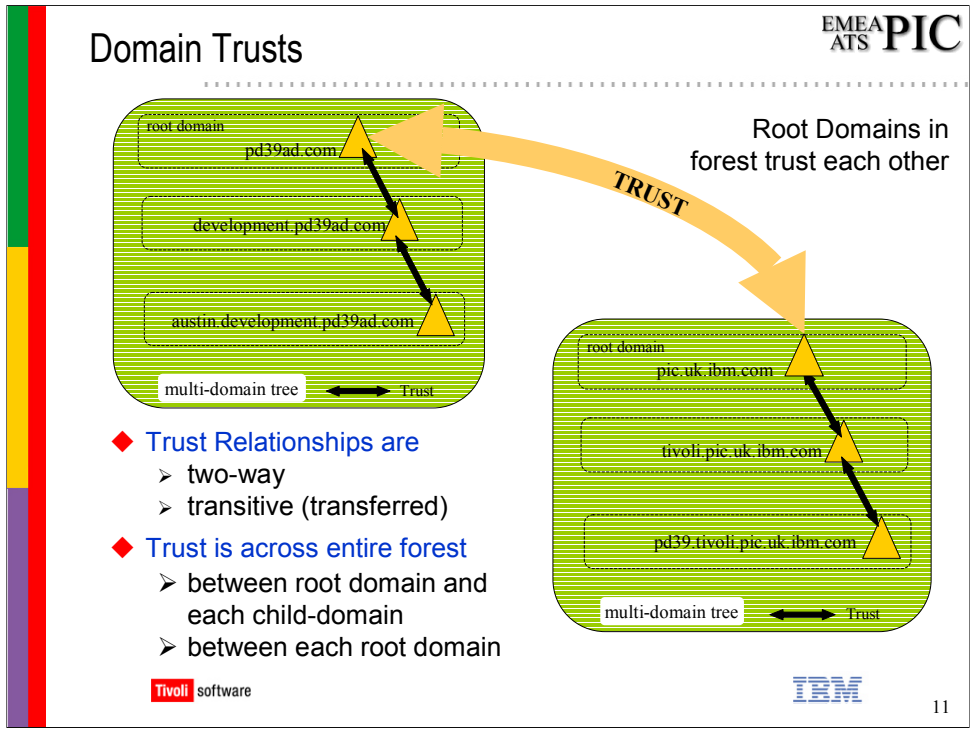



A *domain* is a single security boundary of a Windows 2000 computer network. Every Active Directory contains at least one domain. On a standalone workstation, the domain is the computer itself. A domain can span more than one physical location. Every domain has its own security policies and security relationships with other domains.

A *domain tree* (tree) is comprised of several domains that share a contiguous name space and have a common schema and configuration. Domains in a tree are also linked together by trust relationships. An Active Directory may contain one or more domain trees.

A *forest* is a set of one or more trees that *do not* form a contiguous namespace. All trees in a forest share a common schema, configuration, and global catalog. Unlike trees, a forest does not need a distinct name.

The *forest root domain* is the first domain created in the forest.



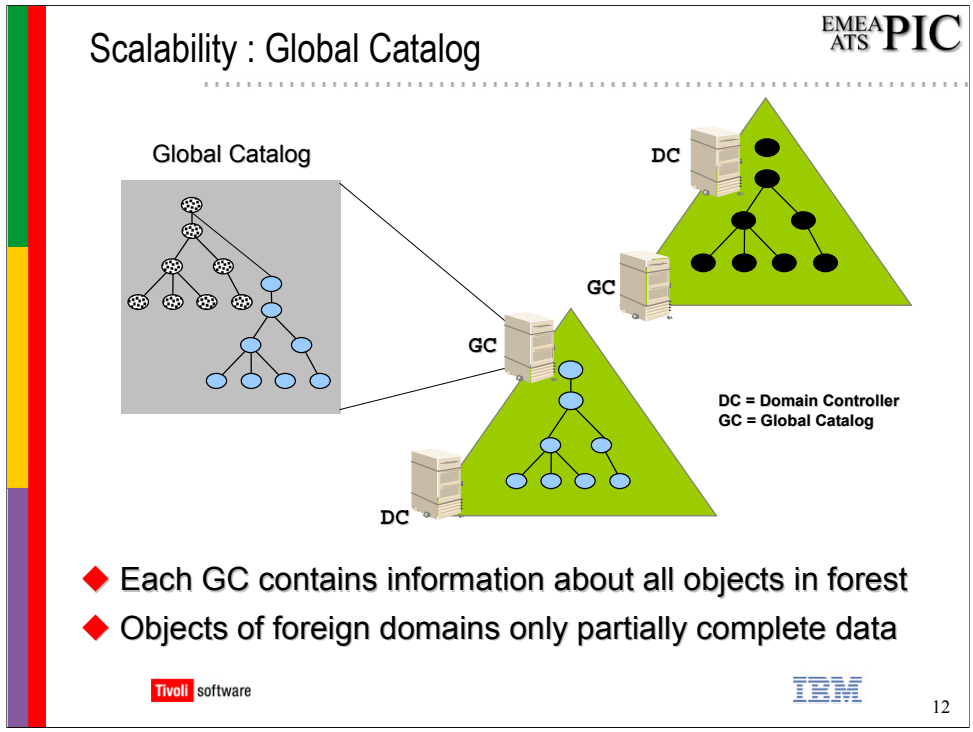

Windows 2000 domains in a tree are joined together through two-way, transitive trust relationships. Because these trust relationships are two-way and *transitive* (ie, trust is transferred from one domain to another) a Windows 2000 domain newly created in a domain tree or forest immediately has trust relationships established with every other Windows 2000 domain in the domain tree.

However, a forest does have a root domain. The forest root domain is the first domain created in the forest. The root domains of all domain trees in the forest establish transitive trust relationships with the forest root domain. In the illustration, *pd39ad.com* is the forest root domain. The root domains of the other domain tree, *pic.uk.ibm.com* has a transitive trust relationships with *pd39ad.com*.

In a forest, a trust relationship is automatically created between the forest root domain and the root domain of each domain tree added to the forest. Because these trust relationships are transitive, users and computers can be authenticated between any domains in the domain tree or forest

The effect of this trust is that:

- users can be authenticated by any domain in forest
- user's credentials are useable by all domains in forest




The Global Catalog is a namespace that contains directory information from all domains in a forest.

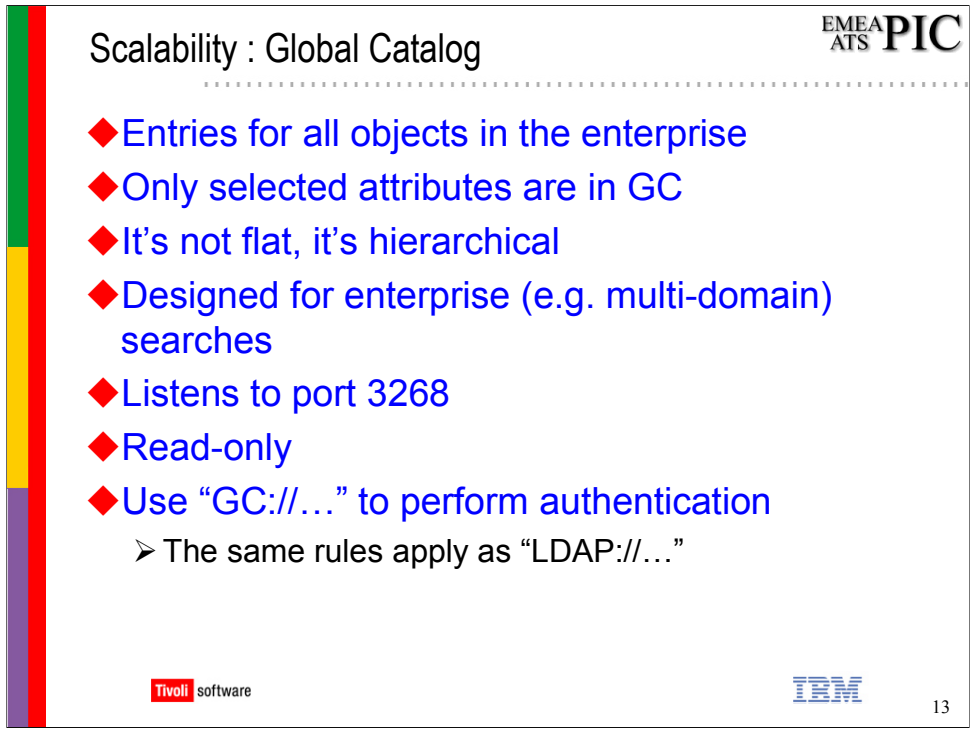
The GC holds a replica of every object in Active Directory but with only a small number of their attributes.

If there is only one domain controller in the domain, the domain controller and the Global Catalog are the same server. If there are multiple domain controllers in the network, the Global Catalog is hosted on the domain controller configured as such.

The attributes in the GC are those most frequently used in search operations (such as a user's first and last names or login names) and those required to locate a full replica of the object.

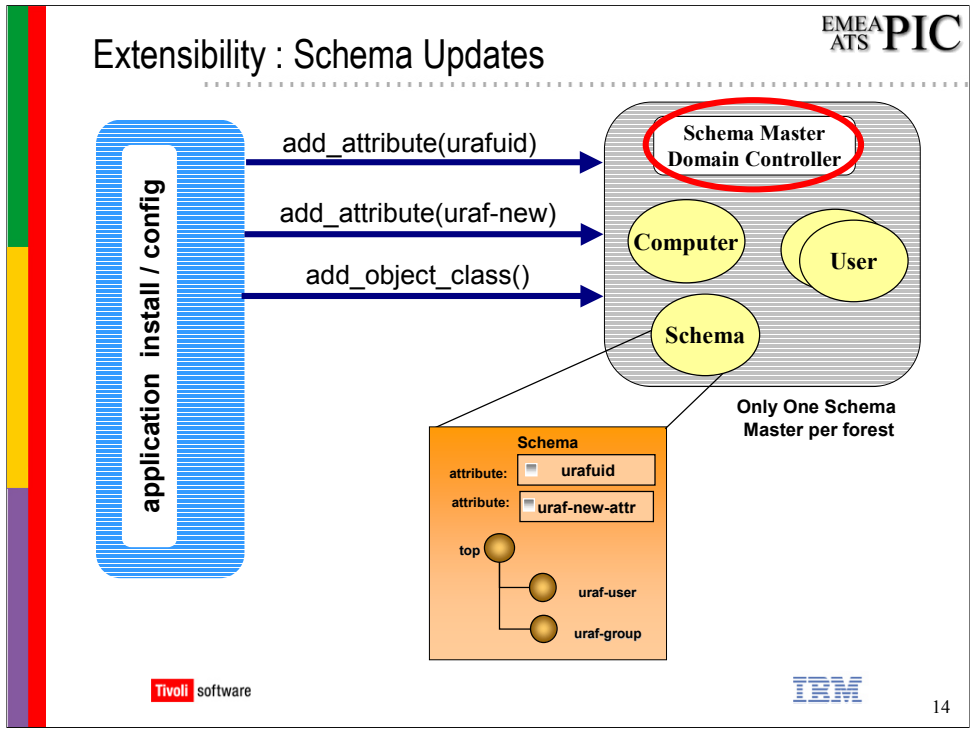
Global Catalogs are designed to respond to user and programmatic queries about objects anywhere in the forest with maximum speed and minimum network traffic. Because a single global catalog contains information about objects in all domains in the forest, a query about an object can be resolved by a global catalog in the domain in which the query is initiated. Thus, finding information in the directory does not produce unnecessary query traffic across domain boundaries.

You can optionally configure any domain controller to host a GC based on the requirements for servicing logon requests and search queries.




The global catalog is built automatically by Active Directory replication system. The replication topology for the global catalog is generated automatically. The properties replicated into the global catalog include a base set defined by Microsoft. Administrators can specify additional properties to meet the needs of their installation.

The Global Catalog is kept on specific servers throughout the enterprise. Only domain controllers can serve as Global Catalog servers. Administrators indicate whether a given domain controller holds a Global Catalog by using the Active Directory Sites and Services Manager.




Active Directory supports schema extensions

- new attributes
- new object classes

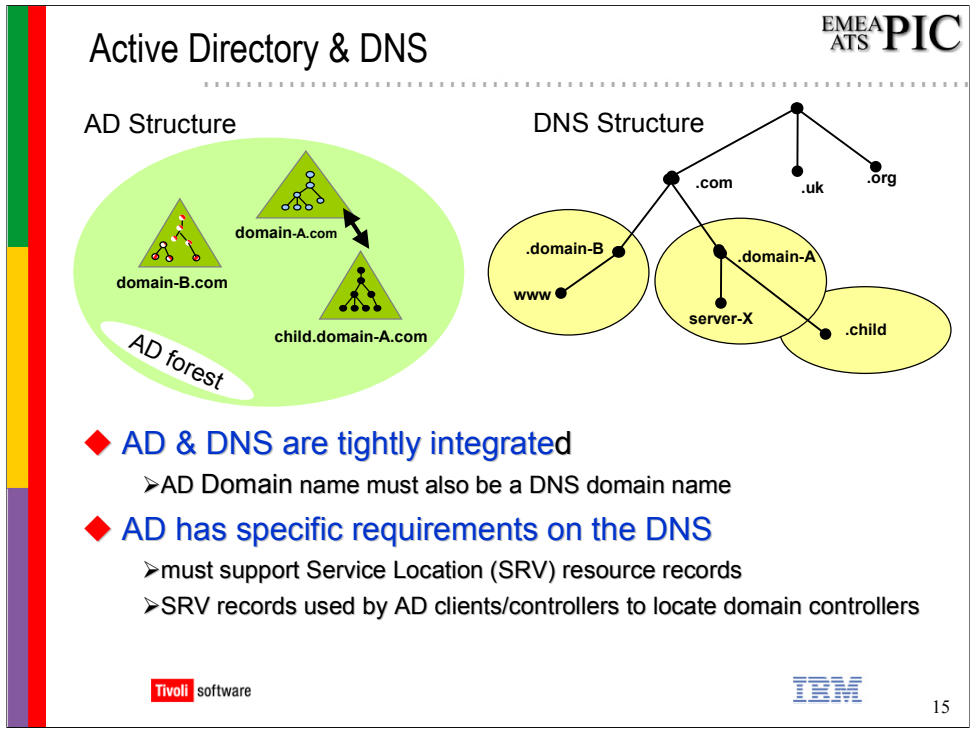
Modifications to schema must be made to ***schema master*** domain controller

- The schema master domain controller controls all updates and modifications to the schema. To update the schema of a forest, you must have access to the schema master. At any time, there can be only one schema master in the entire forest.

start Admin Tools Console with 'mmc' at command window.

Only one domain controller within the forest is the ***schema master.*** All domains within a forest share the same schema.

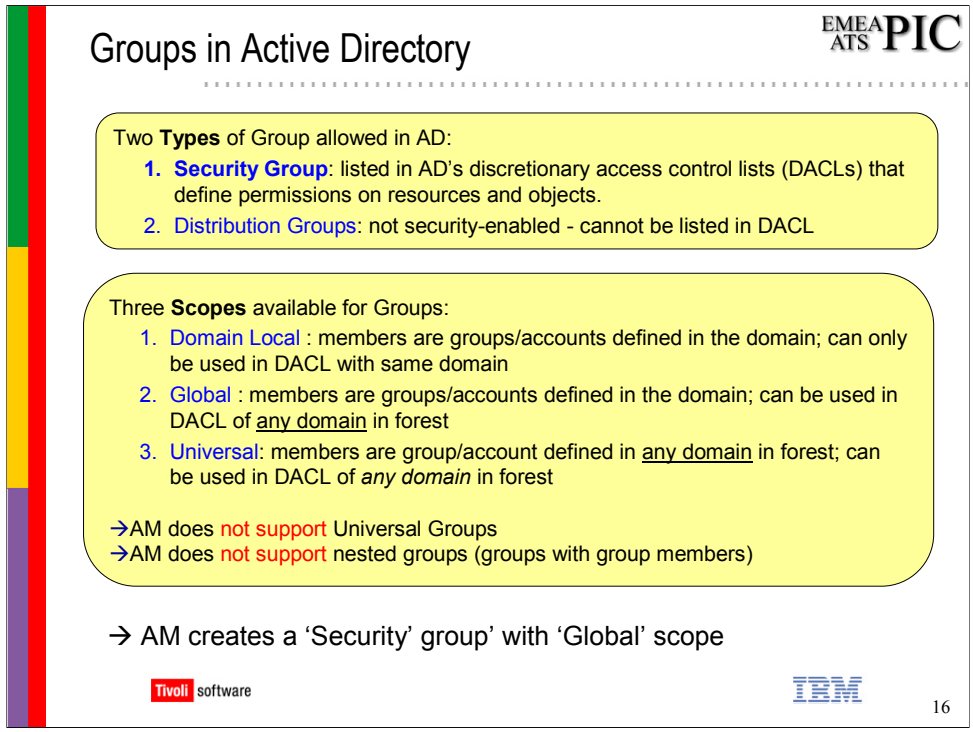
To determine the current ***schema master*** you must use the Schema Management MMC plug-in. (see 'Administrative Tools' later in presentation). Right click on "Active Directory Schema" and select "Operations Master…"




**To view/verify the SRV registration for domain controllers:**

1. **nslookup**
2. Change query type with "**set q=<type>**, where <type> iis the resource record (RR) type to apply as a filter for subsequent lookups.
3. For example, in this instance, because you want to limit subsequent name queries to only return service location (SRV) <types>, issue '**set q=srv**.'
4. Then enter: **_ldap._tcp.dc._msdcs.<domain_name>** where <domain_name> is the DNS name configured for use with your Active Directory domain and any of its associated domain controllers.
5. For example, if the DNS domain name of your Active Directory domain is "pd39ad.com", enter, "**_ldap._tcp.dc._msdcs.pd39ad.com"**
6. Review the output of the previous SRV query and determine if further action is needed based on whether the previous query succeeded or failed:

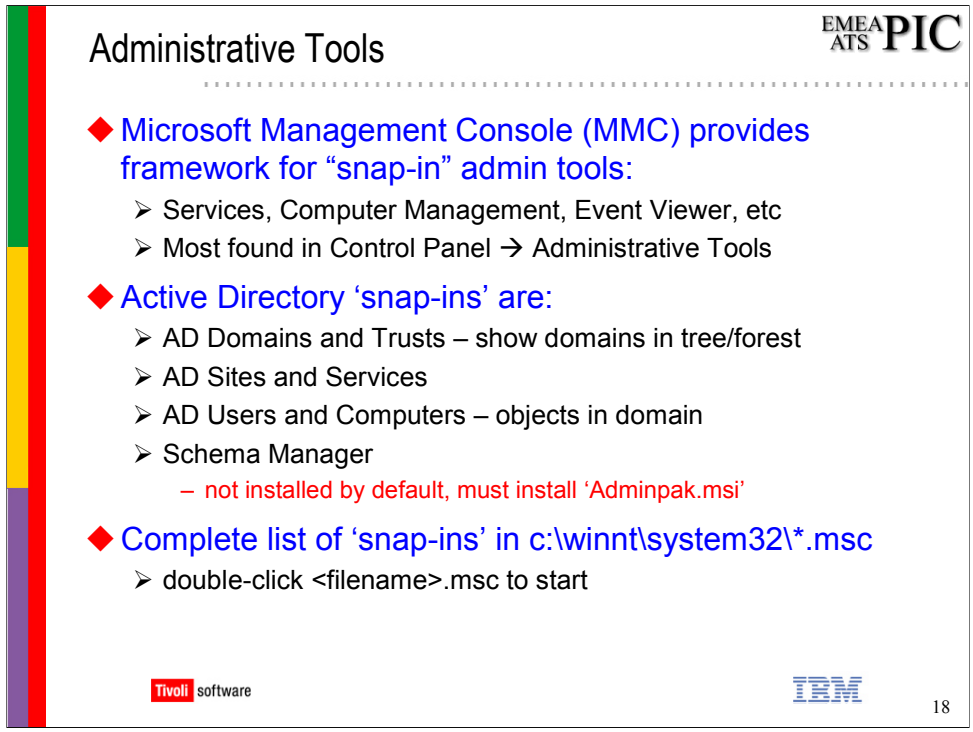
- If the query succeeded, review the registered SRV RRs returned in the query to determine if all domain controllers for your Active Directory domain are included and registered using valid IP addresses.
- If the query failed, continue troubleshooting dynamic update or DNS server related issues to determine the exact cause of the problem.




The groups created by Access Manager are Security groups of Global scope.

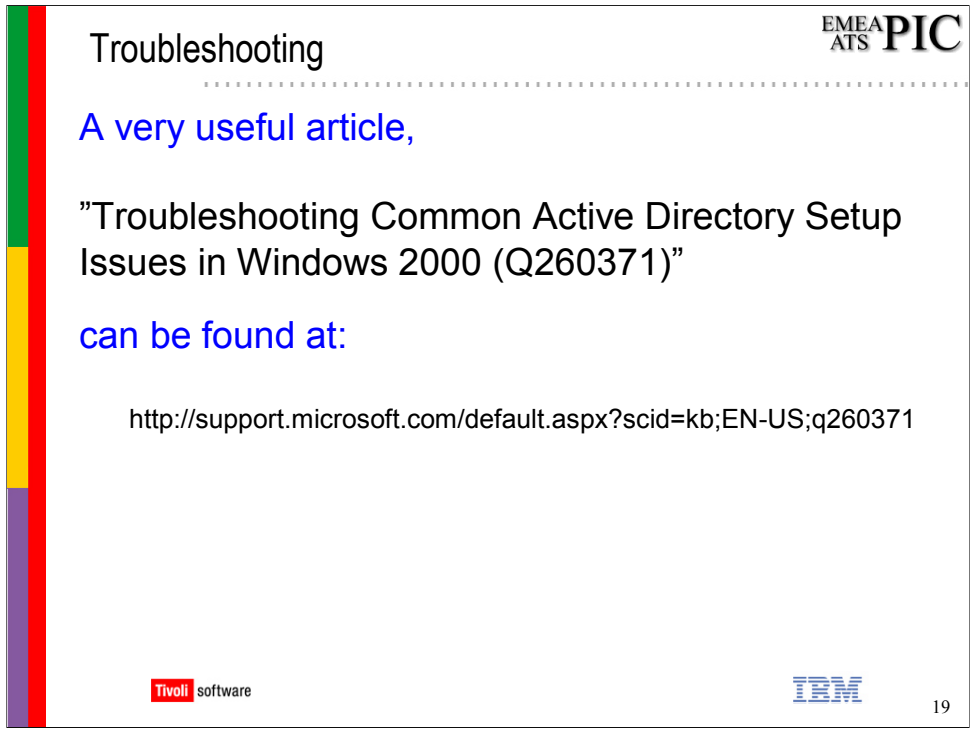
Universal Groups and Nested Groups (groups with group members) are not supported in mixed-mode (domains that contain Windows NT domain controllers). Mixed-mode is the default configuration setting for AD.


EMEA ATS PIC
Active Directory Tools & API
Tivoli software
IBM
17

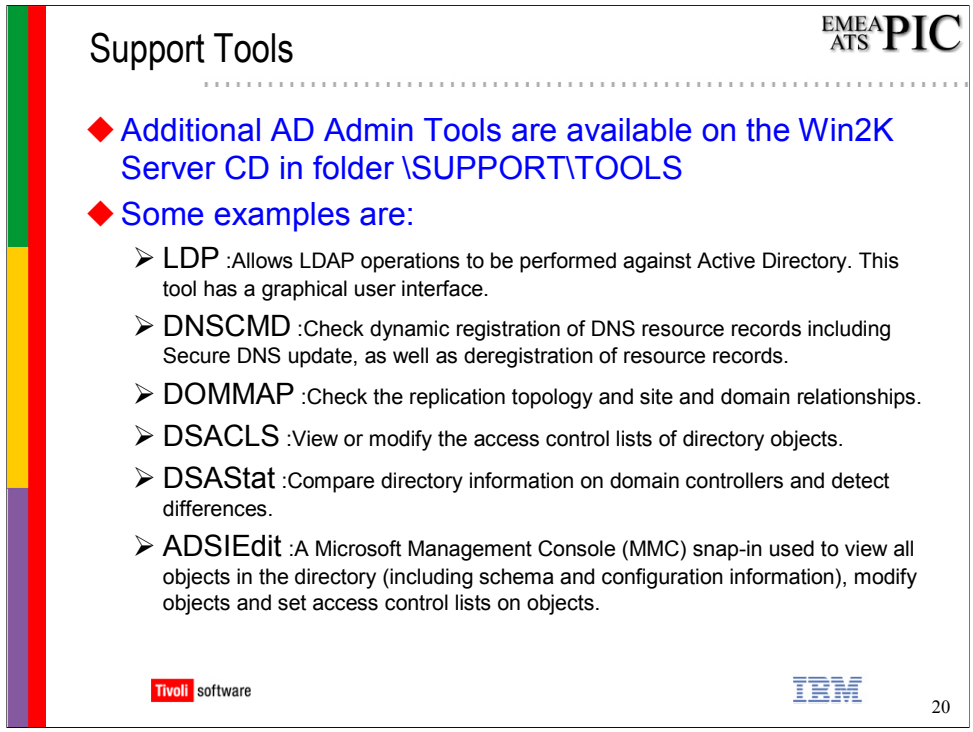



To install the Adminpak.msi that contains the Schema Management snap-in

1. Log on as Administrator.
2. Insert the Windows 2000 Server compact disc into your compact disc drive, and then click **Browse this CD**.
3. Double-click the **I386** folder, double-click , and then follow the instructions that appear in the Windows 2000 Administration Tools Setup wizard.
4. Click **Start**, click **Run**, type **mmc /a**, and then click **OK**.
5. On the **Console** menu, click **Add/Remove Snap-in**, and then click **Add**.
6. Under **Snap-in**, double-click **Active Directory Schema**, and then click **Close**.
7. If you have no more snap-ins to add to the console, click **OK**.
8. To save this console, on the **Console** menu, click **Save**.




An example of a problem documented here is that “File and Printer Sharing” is required on the AD Server to allow a machine to join the AD Domain. Joining the AD Domain is a required part of AM 3.9 configuration.




Utilities within the \SUPPORT\TOOLS package are:

•MoveTree : Move objects from one domain to another.

•SIDWalker :Set the access control lists on objects previously owned by accounts that were moved, orphaned, or deleted.

•NTDSUtil : Repair, check, compact, move, and dump the directory database files. List site, domains and server information, manage operations masters, perform authoritative restore, create domains.

•LDP :Allows LDAP operations to be performed against Active Directory. This tool has a graphical user interface.

•DNSCMD :Check dynamic registration of DNS resource records including Secure DNS update, as well as deregistration of resource records.

•DOMMAP :Check the replication topology and site and domain relationships.

•DSACLS :View or modify the access control lists of directory objects.

•NETDOM5 :Batch management of trusts, joining computers to domains, verifying trusts and secure channels.

•NETTest :Check end to end network and distributed services functions.

•NLTest :Check that the locator and secure channel are functioning.

•REPAdmin :Check replication consistency between replication partners, monitor replication status, display replication metadata, force replication events and knowledge consistency checker recalculation.

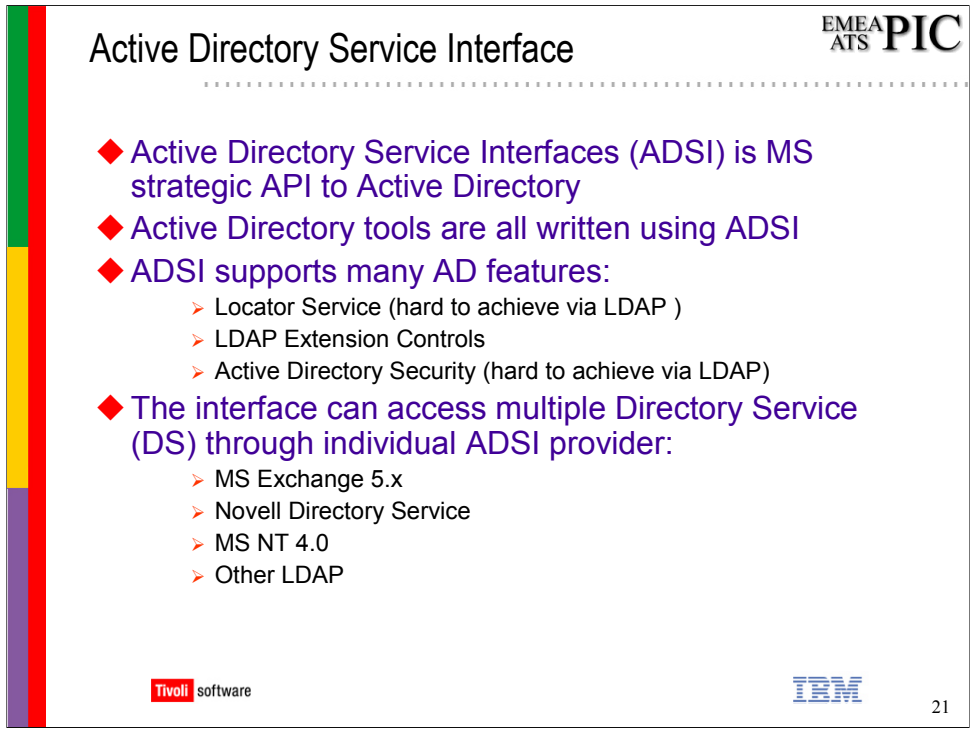
•REPLMon :Display replication topology, monitor replication status (including group policies), force replication events and knowledge consistency checker recalculation. This tool has a graphical user interface.

•DSAStat :Compare directory information on domain controllers and detect differences.

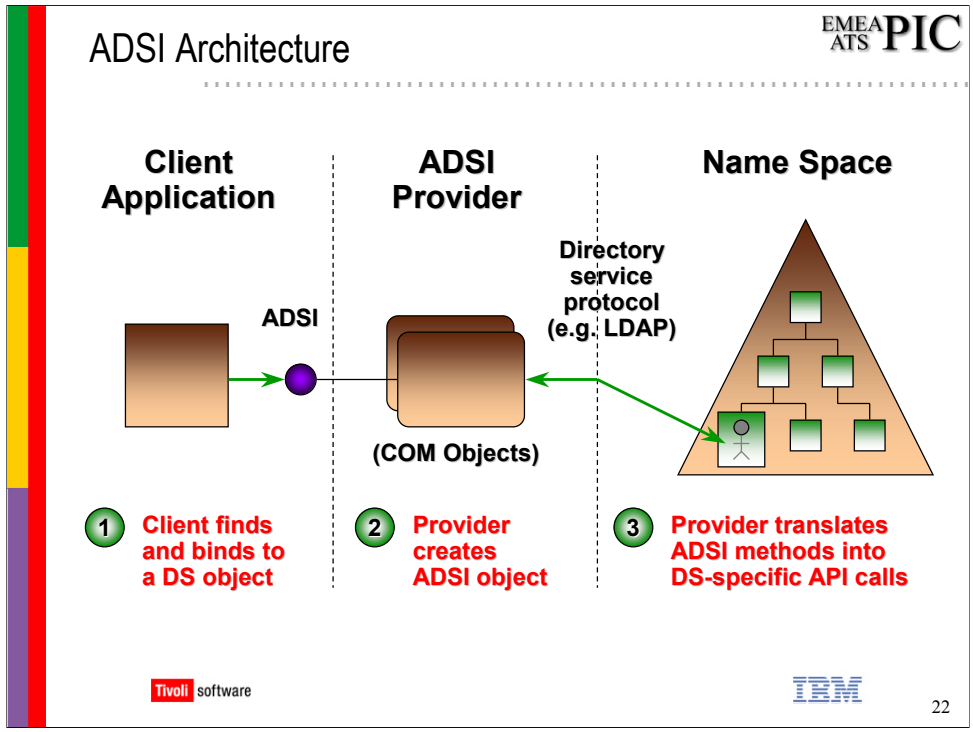
•ADSIEdit :A Microsoft Management Console (MMC) snap-in used to view all objects in the directory (including schema and configuration information), modify objects and set access control lists on objects.

•ACLDiag :Determine whether a user has been granted or denied access to a directory object. It can also be used to reset access control lists to their default state.

•DFSCheck :Command line utility for managing all aspects of Distributed File System (DFS), checking the configuration concurrency of DFS servers, and displaying the DFS topology




Active Directory Service Interface (ADSI) only supported on Windows 2000 platforms. An ADSI client (Version 2.5) is also provided by MS from its web site for ADSI support on Windows NT platform. However, as current understanding, its ADSI functionalities may not be equivalent as provided on Windows 2000 platforms.




An ADSI provider contains the implementation of ADSI objects and dependent objects for a particular namespace.

ADSI Services include:

- Bind to a DS object
- Enumerate objects within a DS object
- Read/write properties (attributes) from/to a DS object
- Manage the schema of a DS
- Manage security on objects
- Query a DS and return result sets


EMEA ATS PIC
Access Manager & Active Directory
Tivoli software
IBM
23

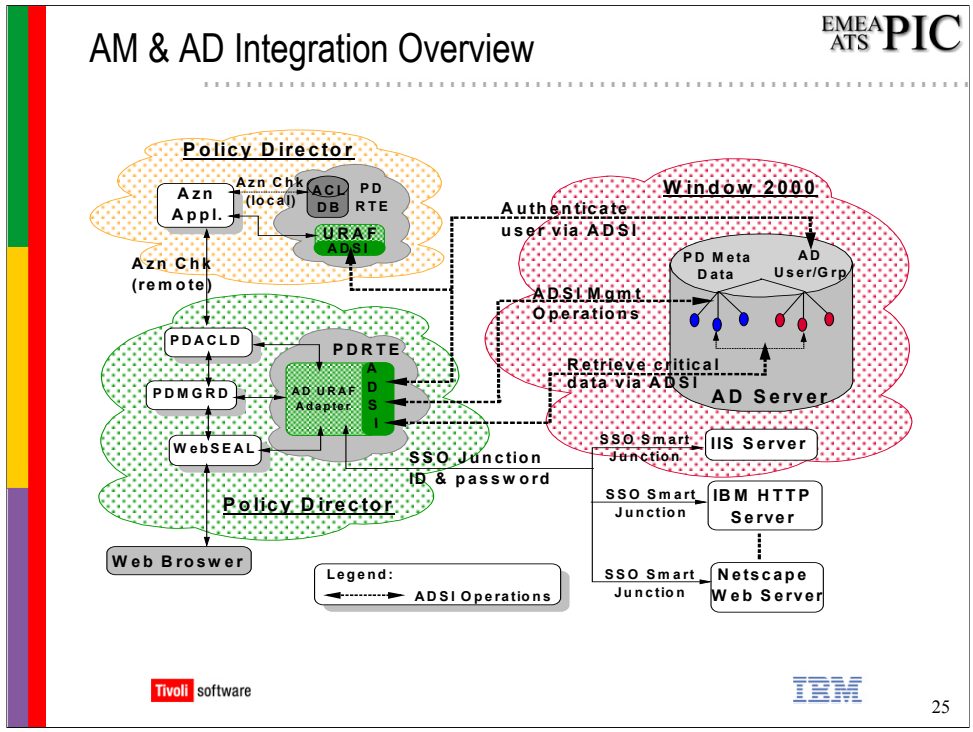
## Access Manager AD Support : Overview

EMEA ATS PIC

- Windows 2000 Advanced Server
- Two modes of configuration:
  - **single-domain** : all users/groups in single AD Domain
  - **multi-domain** (single forest)
    - users can be defined in any domain
    - group & users must be in the **same domain** (Universal group not supported in this release)
    - a Global Catalog **must** be available in forest
- Failover Support
  - **single-domain** – fail-over support to multiple domain controllers
  - **multi-domain** :
    - failover support to multiple domain controllers in forest root only
    - when domain controller in non-root domain fails, other domains continue to operate
  - no “prefer replica” support – failover is to any available domain controller

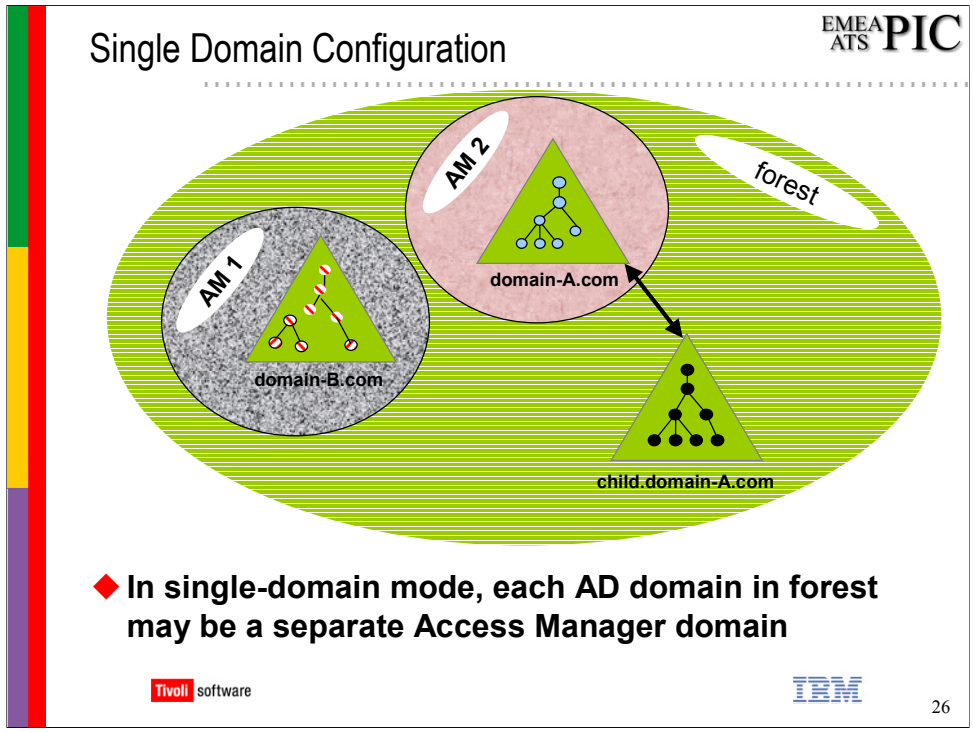
Tivoli software IBM

24

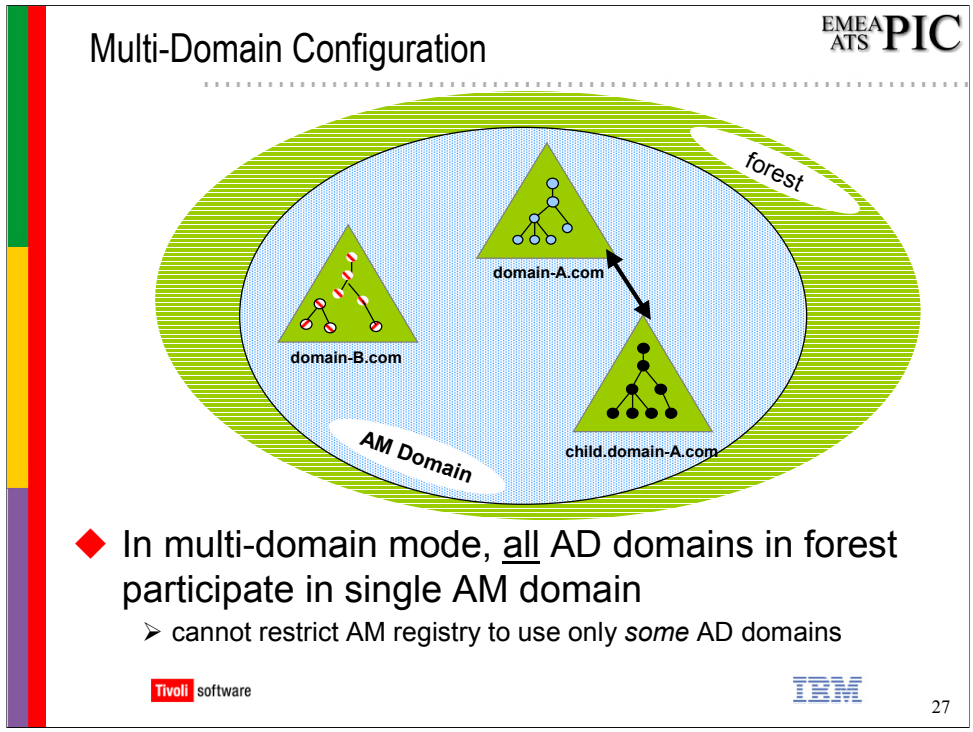



Access Manager is using the User Registry Adapter Framework (URAF) design to achieve the support of integration with Active Directory registry using Active Directory Service Interface (ADSI).

All operations from AM components which associated with user registry are all going through the Active Directory URAF adapter via URAF interfaces. Under the covers, the URAF adapter is using ADSI to communicate with Active Directory and manage all the URAF data objects that have tied with Access Manager operations. From Access Manager point of view, even though the underneath user registry has been changed to Active Directory, all the regular operations will remain the same as in other registries (e.g. LDAP).

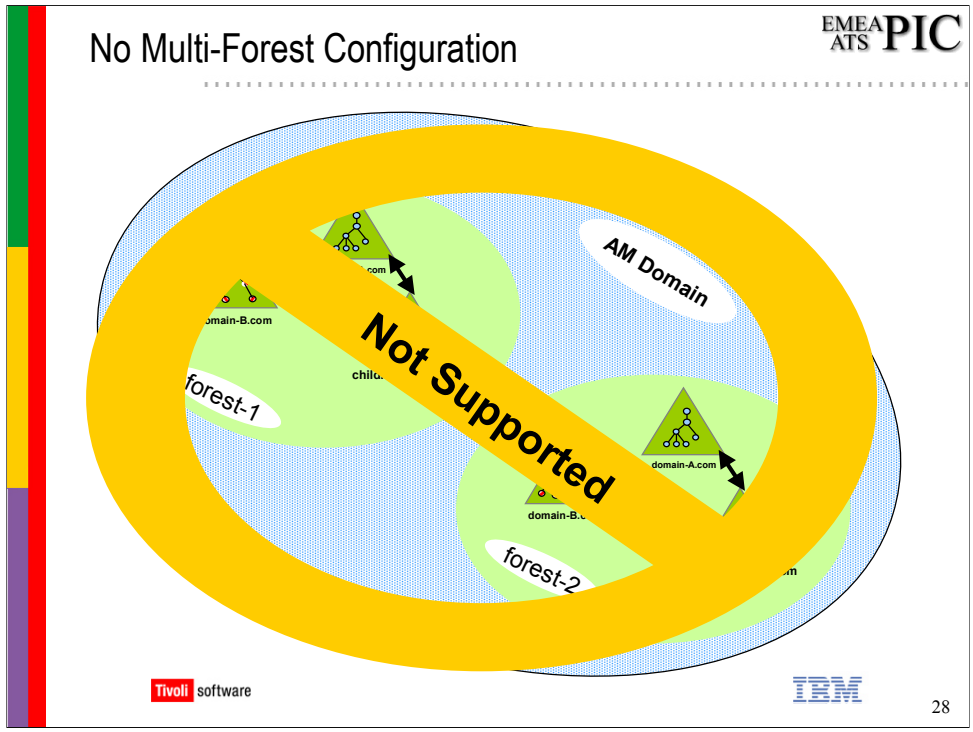


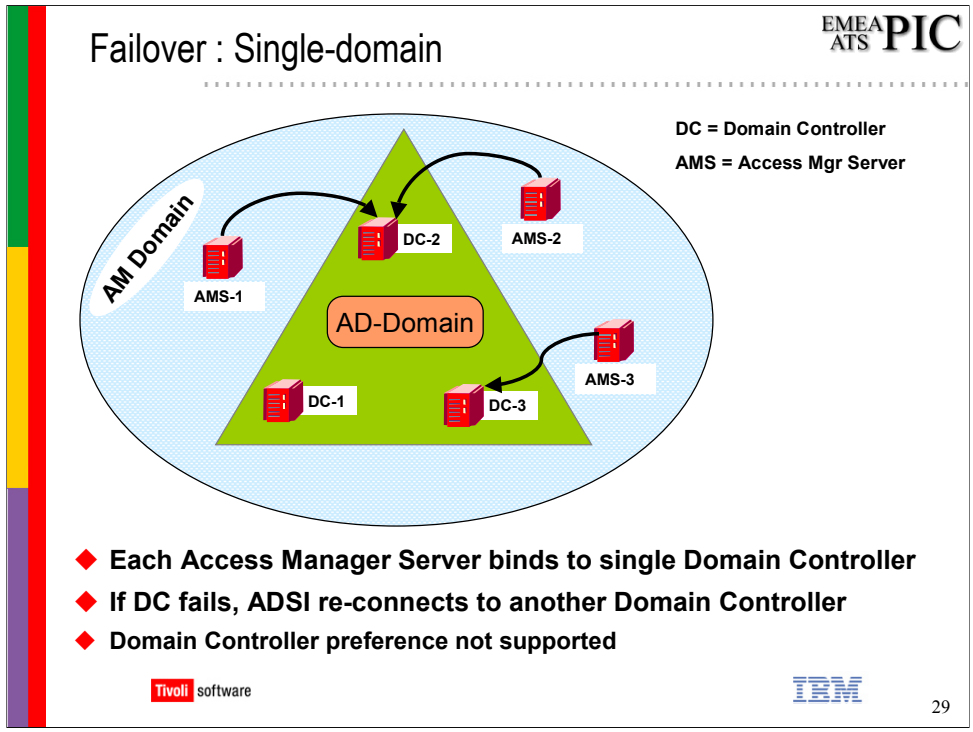

When configured for “single-domain”, all Access Manager components use a single AD Domain to contain user and group definitions.




In this mode, users and groups from all AD domain can participate in the AM Domain.

A user's group membership, however, is restricted to groups within the domain.


No Multi-Forest Configuration
EMEA ATS PIC
AM Domain
Not Supported
forest-1
forest-2
domain-A.com
Tivoli software
IBM
28




- Access Manger uses *serverless binding* to handle failover support. This means that AM does not specify a Domain Controller (DC) when binding to AD. The ADSI uses the locater service to choose a DC for binding.

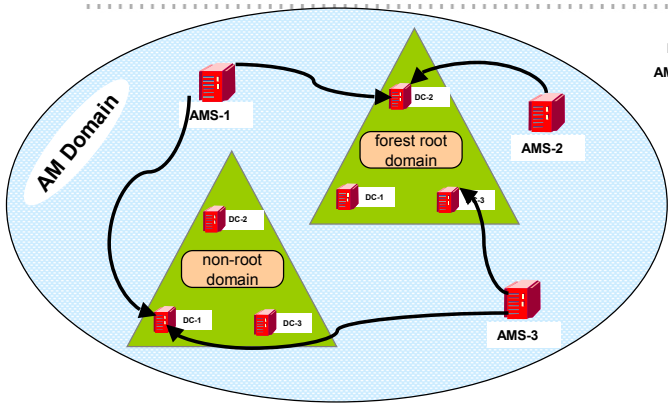
This is possible, because AD supports multi-master replication so all DCs can be used for updates. Note that the data propagation between Domain Controllers is time intensive, and may cause delays until an update on one DC is available on other DCs in the domain.

When the current DC fails, the ADSI automatically chooses another DC in the domain.

It is not possible to specify a preference for the DC chosen at initial bind or for failover.

# Failover : Multi-domain

EMEA ATS PIC

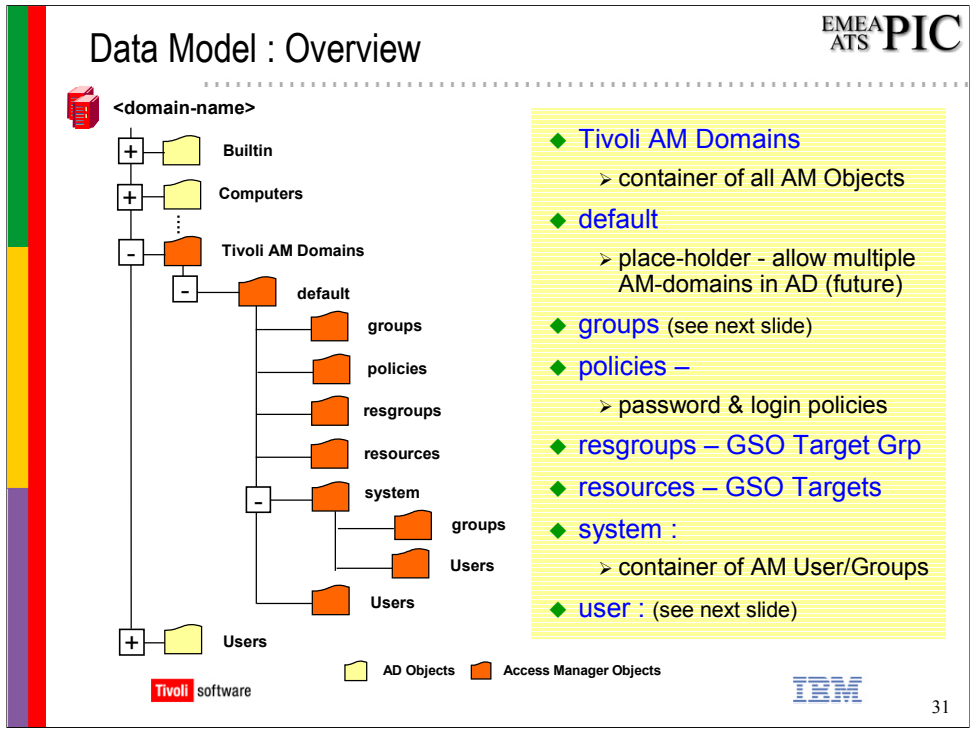



- Each Access Manager Server binds to single Domain Controller on each domain
- If Dom. Controller fails ADSI automatically re-connects to another DC
  - *only on forest root domain,* not supported on non-forest root dom.
- DC preference not supported

Tivoli software

IBM

30




**Note: To see the Access Manager objects you must enable the "Advanced" view on the Active Directory's "Users and Computers" MMC console.!!!**

The "Tivoli AM Domains" sub-tree created under the primary Window 2000 domain's DN contains all the data objects related to Access Manager. Under the "default" AM domain container, these containers partition the name space for groups, policies, resgroups, resources, and users.

ivmgrd-master is member of:

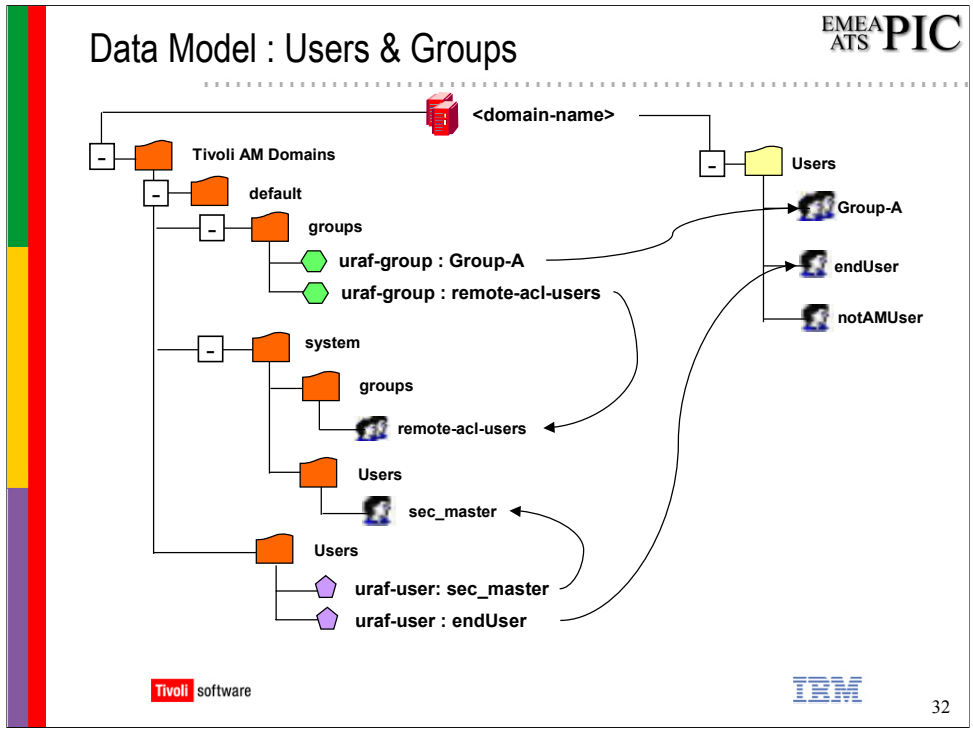
> Administrators, Domain Admins, Domain Users, ivacld-servers, ivmgrd-servers, secmgrd-servers, SecurityGroup

sec_master is member of:

> Administrators, Domain Admins, Domain Users, ivacld-servers, iv-admin, ivmgrd-servers, remote-acl-users, secmgrd-servers, SecurityGroup

webseald-secureway1 is member of

> Domain Users, ivacld-servers, secmgrd-severs, SecurityGroup, webseal-servers




Active Directory represents a user with a User object, and a group with a Group object.

Access Manager represents a user with a URAFf-User object, and a group with a URAF-Group object.

A **uraf-user** object contains a userID, UUID and registryUID that 'points' to a User object.

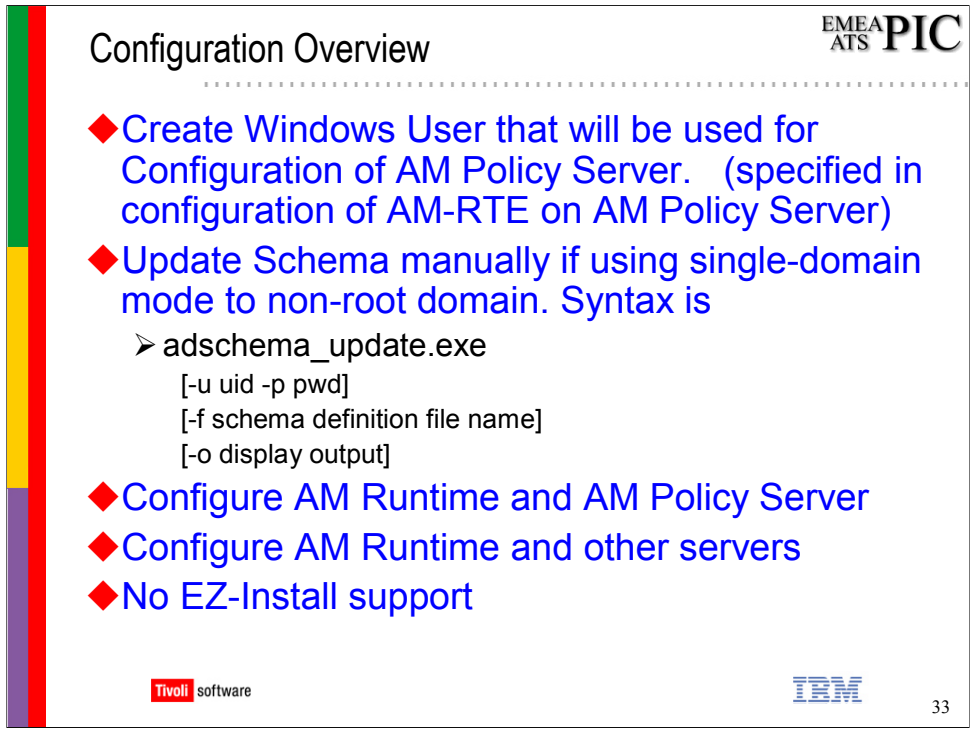
A **uraf-group** object contains a groupID, UUID, and registryGID that 'points' to a Group object.

For pdadmin> group & user *import*, the URAF-Group or URAF-User object is created that points to the existing object.

For pdadmin> group or user *create*, both URAF-Group or URAF-User object and the corresponding User/Group object is created. Following AD convention the User or Group object is created under cn=User,dc=<domain>.

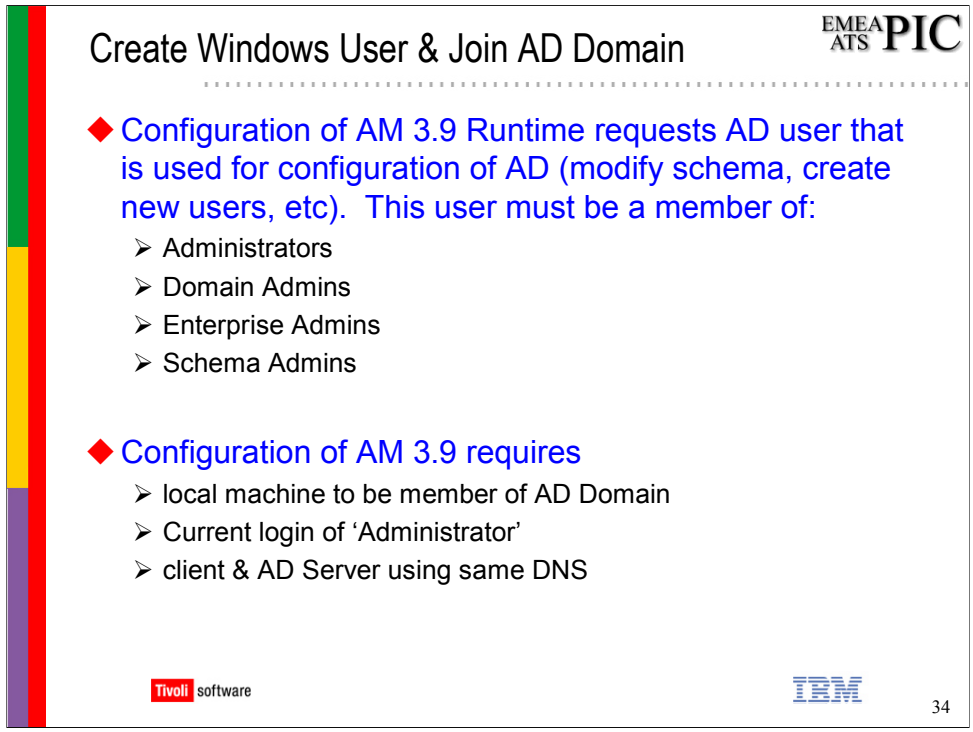
It is the UUID that is used in Access Manager ACL to represent a User or Group.

Access Manager creates several Users and Groups for use with the root user sec_master and AM servers and applications. These are created in under cn=system,cn=default,dc=Tivoli PD Domains,dc=<domain>




The schema definition file name is <AM>\Policy Director\etc\adschema.def.

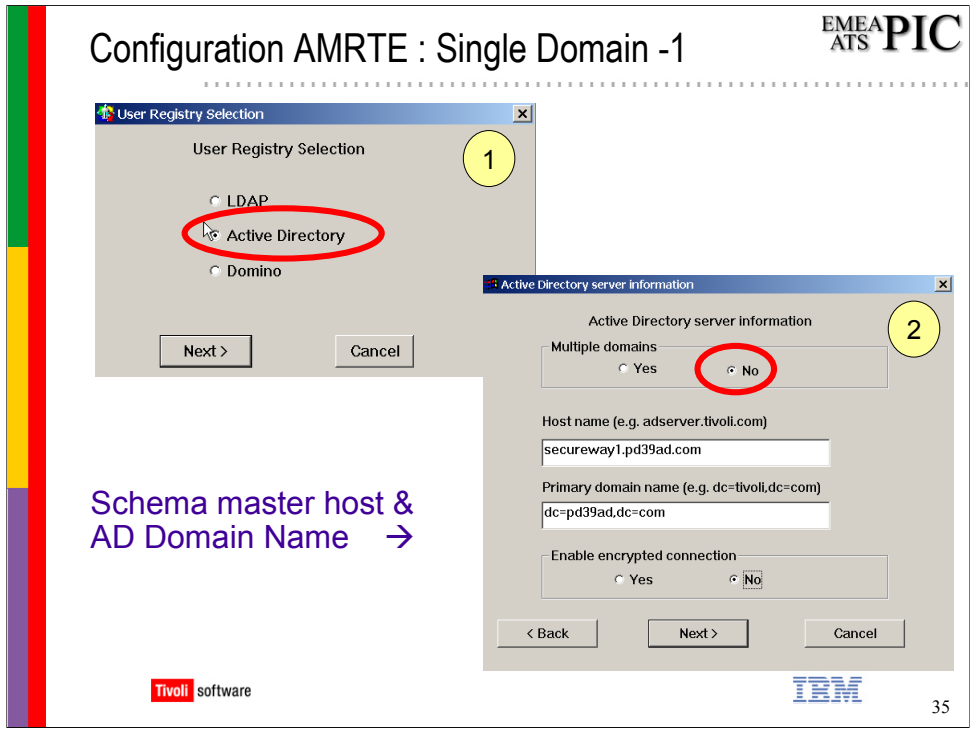
Note: For Multi-domain configuration all the *system entities* (user & groups) that represent Access Manager servers, will be created in the forest root domain.




Access Manager Installation and Configuration is covered in the Beta versions of the AM Publications.

The schema definition file name is <AM>\Policy Director\etc\adschema.def.

Note: For Multi-domain configuration all the *system entities* (user & groups) that represent Access Manager servers, will be created in the forest root domain.




This slide and the next show the dialogues that are displayed when configuring Access Manager Runtime Environment for use of a single AD domain. These slides show the dialogues displayed for a machine *with* the AM Policy Server installed. When configuring AM-RTE on a machine that does not have AM Policy Svr installed you will be prompted to enter the host name of the AM Policy Sever machine.

1. Beginning with AM 3.9, Active Directory is now a choice on the User Registry Selection panel.

2. This dialogue provides configuration information about the AD environment.

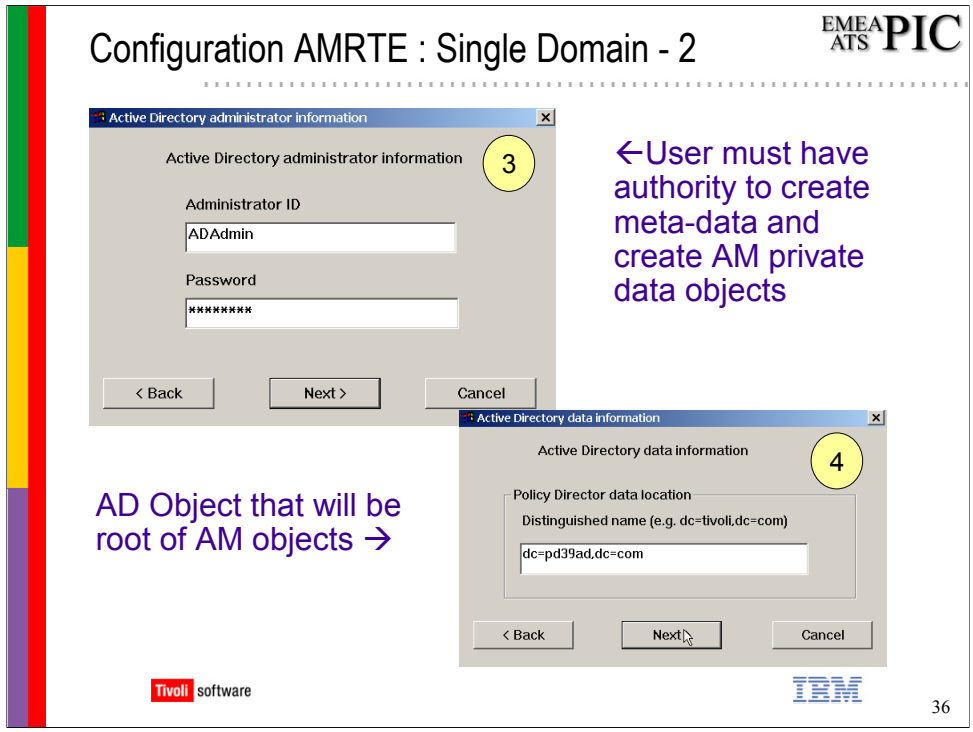
-To configure AM to use a single AD domain, choose Multi-domains -> NO.

-The host name must be the schema master domain controller. AM modifies the schema to create object classes that are specific to AM. At any time, there is only only one schema master in the entire forest.

> **Note:** if the domain your are using is not a root domain, you must first manually update the schema using adschema_update.exe

- Enter the name of AD Domain.

- To enable an encrypted connection between AM and the AD servers choose 'Yes'. AM uses Kerberos and encryption services of AD to communicate with the AD Domain Controllers.



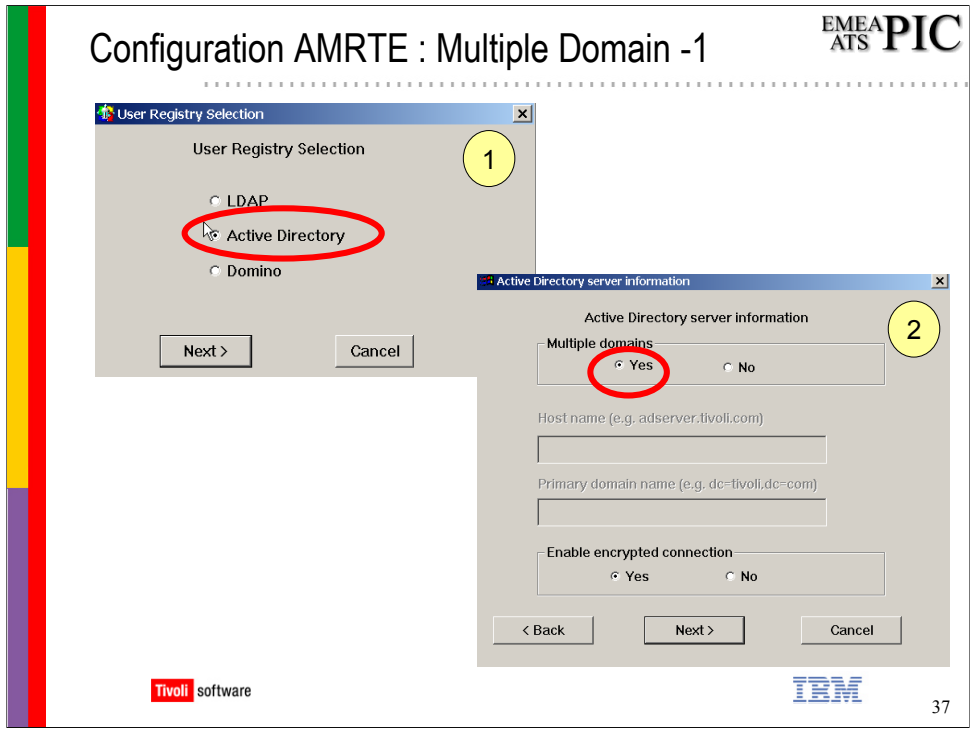

As usual, after configuration completes, check <AM Install>\Access Manager\log\config.log to see activity performed during configuration.

In fact, it is possible to make a mistake in configuration of AMRTE and the configuration be successful! This is because the information entered at AMRTE configuration may not be used until configuration of an AM Server.

If configuration of an AM Server fails, you my want to un-configure both the Server and the Runtime Environment.

After PDRTE configuration, these files should be in <AM Install>\Access Manager\etc:

- AM.conf
- activedir.conf
- ldap.conf (not used for AD)



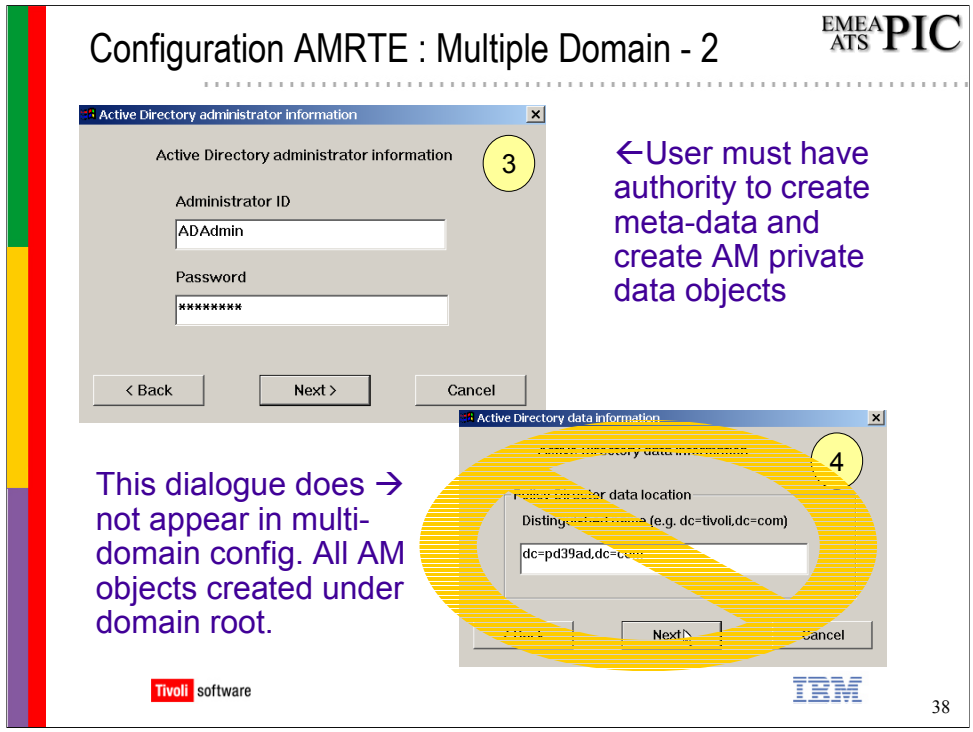

This slide and the next show the dialogues that are displayed when configuring Access Manager Runtime Environment for use of all AD domains with an AD forest. These slides show the dialogues displayed for a machine *with* the AM Policy Server installed. When configuring AM-RTE on a machine that does not have AM Policy Svr installed you will be prompted to enter the host name of the AM Policy Sever machine.

1. Beginning with AM 3.9, Active Directory is now a choice on the User Registry Selection panel.

2. This dialogue provides configuration information about the AD environment.

-To configure AM to use all AD domains, choose Multi-domains -> Yes..

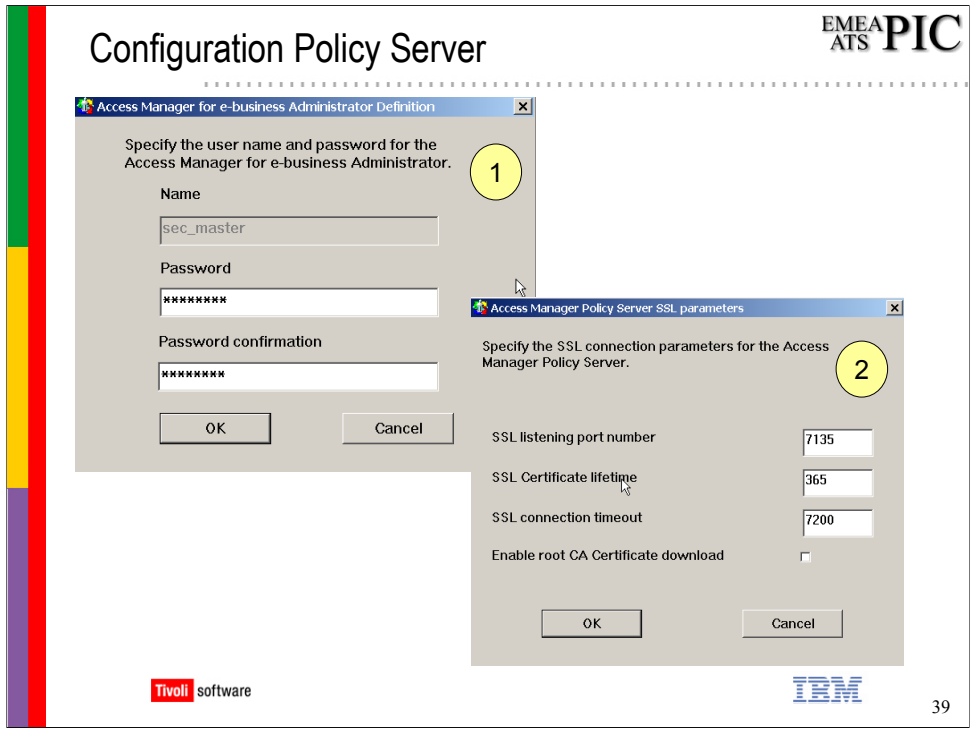
Unlike the single domain configuration, it is not required to enter the name of the schema-master machine or, of course, the name of the domain to be used.




As always, after configuration completes, check <AM Install>\Access Manager\log\config.log to see activity performed during configuration.

After PDRTE configuration, these files should be in <AM Install>\Access Manager\etc:

- AM.conf
- activedr.conf
- ldap.conf (not used for AD)




The configuration of the Policy Server is the same for both single-domain and multiple domain cases.

• creates the ‘sec_master’ user with the given password.

Using the AD Administrator ID, the ‘sec_master’ user is made a member of:

Administrators, Domain Admins, Domain Users, ivacld-servers, iv-admin, ivmgrd-servers, remote-acl-users, secmgrd-servers, SecurityGroup

(2) Configures the usual listening port number, certificate life-time and SSL connect timeout periods for AD Manager. Nothing unique to AD.

There is, however, one significant difference between single-domain and multi-domain case. The ‘sec_master’ User is also a member of the “Enerprise Admins” group. This allows sec_master to create users in all domains in the forest.

Again, please note that not all data entered for AM-RTE is used until configuration of an AM Server. If the configuration of AM Policy Server fails, re-configure AM-RTE and ensure data is correct.

## User & Group : Domain Qualifiers

EMEA ATS PIC

- In multi-domain configuration, specification of user/group requires use of “domain qualifier”
  - example : endUser**@domain.com**
  - example : group**@domain.com**
- Not required when:
  - single domain configuration
  - Operation provides full DN
    - Create of user/group
    - Import of user/group
  - User/Group defined in the forest root domain

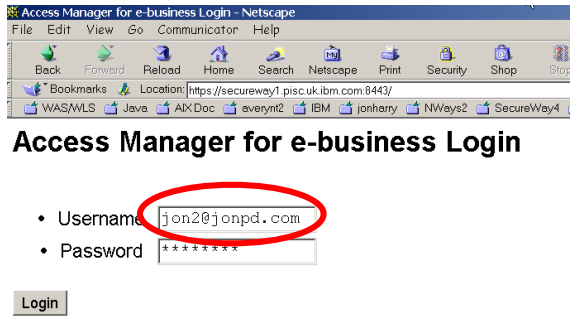
Tivoli software

IBM

40

# User/Group : Domain Qualifiers Examples

EMEA ATS PIC




Tivoli software

IBM

## User/Group : Domain Qualifiers Examples

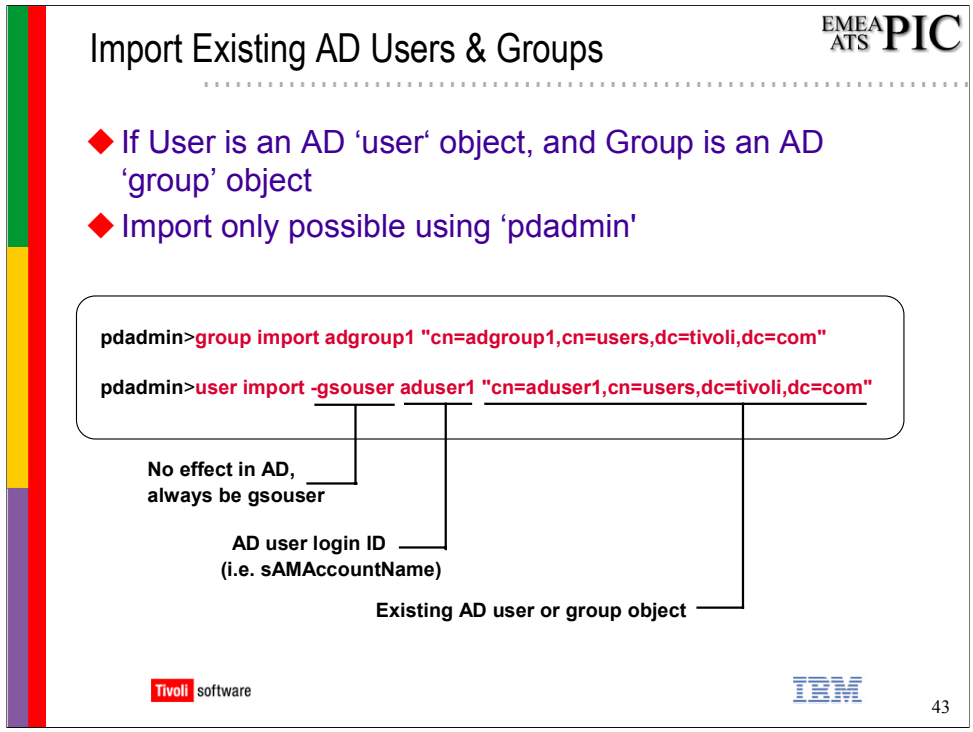
EMEA ATS PIC

```
pdadmin> acl show testACL
      User sec_master@pd39ad.com TcmdbsvaBl
      User ivmgrd-master@pd39ad.com Tr
      User pdPermTest@pd39ad.com Tr
pdadmin> acl modify testACL set user jon2  rT  <<< no domain qualifier
Error: Object not found. (status 0x1712207a)
pdadmin> acl modify testACL set user jon2@jonpd.com rT
pdadmin> acl show testACL
ACL Name: testACL
Description:
   Entries:
User sec_master@pd39ad.com TcmdbsvaBl
User ivmgrd-master@pd39ad.com Tr
User pdPermTest@pd39ad.com Tr
User jon2@jonpd.com Tr
```

Tivoli software

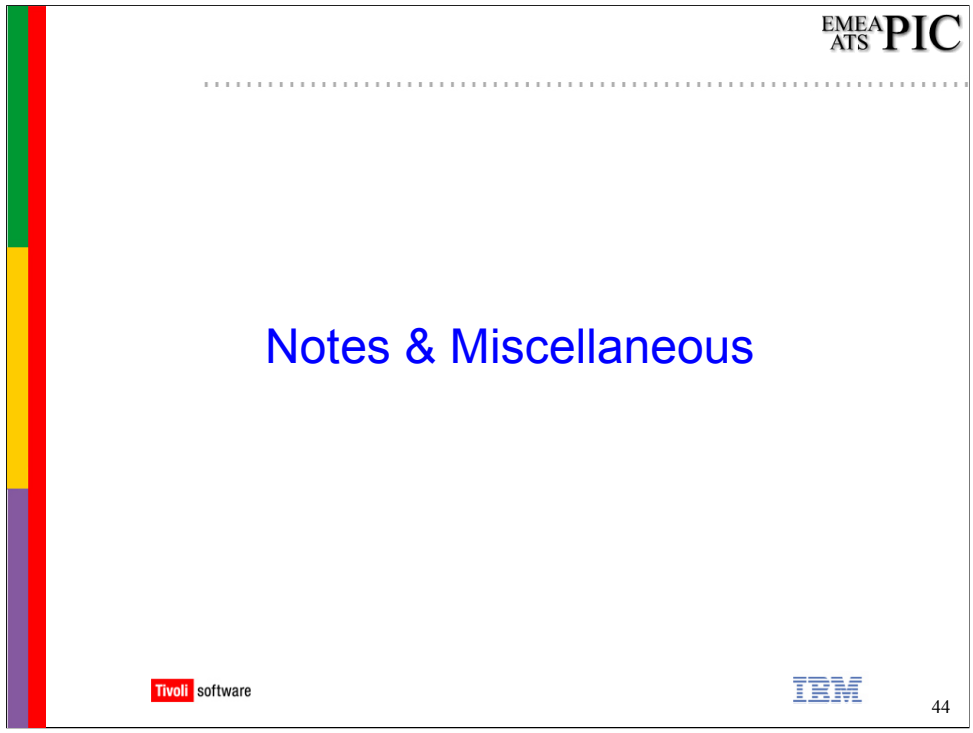
IBM

42




For User or Group *import* the full DN of the existing object provides the domain of the User or Group object so it is not necessary to specify the '@domain' qualifier.

The "-gsouser" has no effect using the AD Registry. All Users are GSO Users and can be assigned GSO Credentials if necessary.


EMEA ATS PIC
Notes & Miscellaneous
Tivoli software
IBM
44

## Notes-1

EMEA ATS PIC

- Easy Install from AM 3.8 cannot work with Active Directory registry
- To import an existing AD user as a AM user, you must use the AD user's login name (i.e. sAMAccountName) as the AM userID
- First 20 characters of User name must be unique
  - limitation of sAMAccountName attribute

Tivoli software

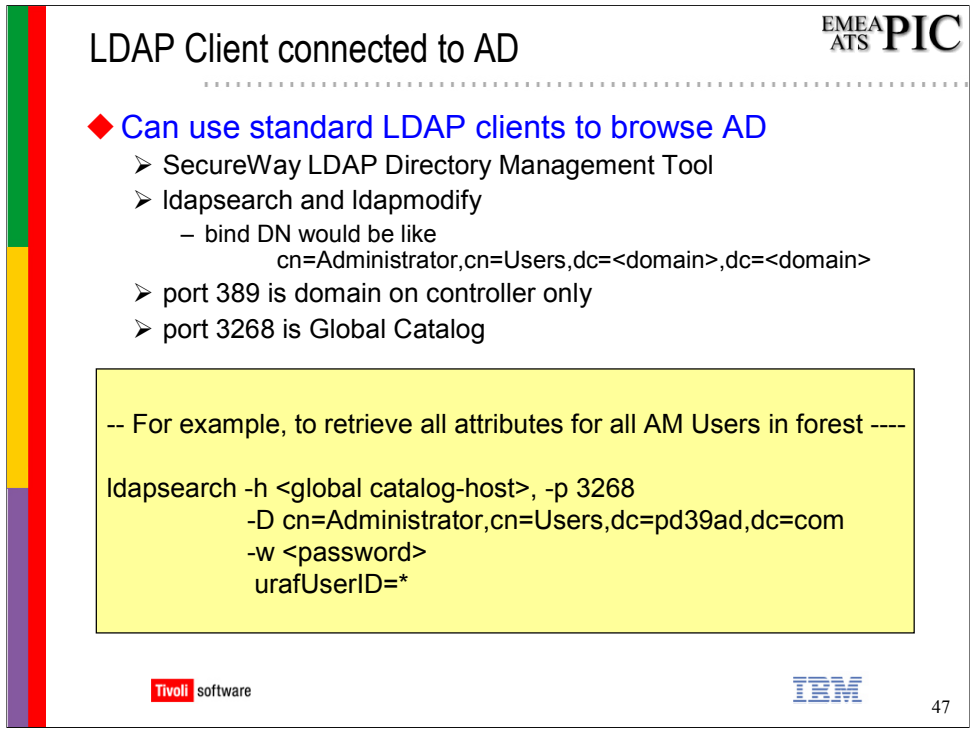
IBM

45

## Notes - 2

EMEA ATS PIC

- dynamic business entitlements (tag/value) data only extracted from LDAP user registry
  - attributes within [ldap-ext-cred-tags] stanza of pd.conf will be placed into credential only if
    - LDAP registry (not AD or Domino)
    - passwd-ldap is authentication mechanism (also 3.8 restriction)
  - use a “credential extended attributes CDAS” (specified via ‘cred-ext-attrs’ within webseald.conf) to provide dynamic data in user credential
    - see WebSEAL Developer Reference for details




This is useful because the standard 'Users & Computers' AD tool will *not* show the attributes of AM objects like URAF-User, URAF-Group.

The 'ADSI Edit' tool will show all attributes of AM Objects.

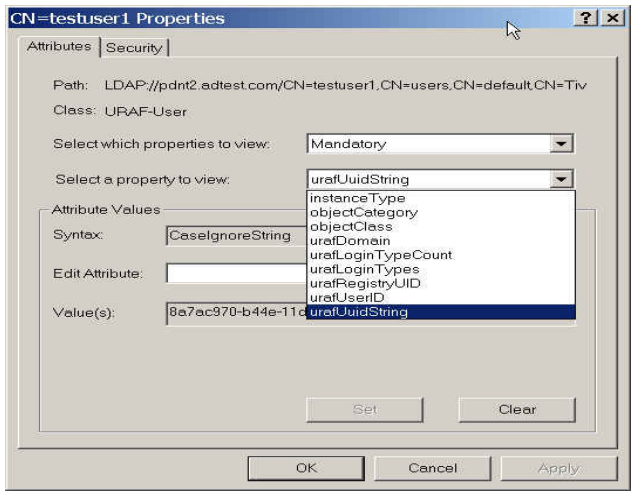
As another example using ldapsearch, to retrieve all the groups in a single domain use:

```
ldapsearch -h <domain-controller>  -p 389
           -D cn=Administrator,cn=Users,dc=pd39ad,dc=com
           -w <pwd>
           -b dc=<domain>,dc=<domain> urafGroupID=*
```

The SW Directory Management Tool does not show all domains when connected to Global Catalog. Probably because it requests 'suffixes' from GC and does not get a response??

## AM Objects in AD : URAF-User Properties

EMEA ATS PIC




Tivoli software

IBM

48

This is an example of the Properties of a URAF-User object as displayed by the “ADSI-Edit” tool – available in the “Support Tools” package on the Win2K Server CD.

You can do the same of URAF-Group objects or any object with AD.

Object attributes can be viewed or edited using this tool.