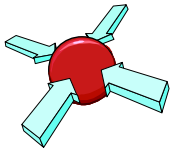
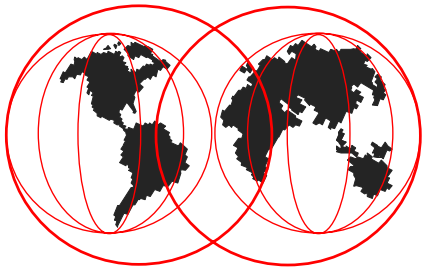
OS/390 Security Server: LDAP



LDAP Overview



Agenda

- What is a Directory Service?
- What is Lightweight Directory Access Protocol (LDAP)?
- What is it used for? (typical applications and benefits)
- What is the LDAP history on OS/390?
- Why is LDAP viewed as a security tool?
  - What is the future of LDAP?
- Implementation examples

What is a directory ?

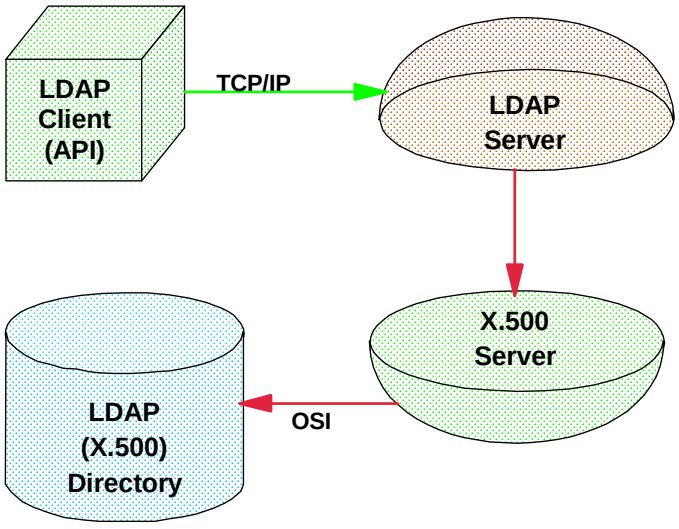
Lightweight Directory Access Protocol

- What is a DIRECTORY?
  - listing of information about objects - phone directory, library card catalog
  - specialized database - read bias, static data, not transaction based (commits), simplified access methods (LDAP) not complex (SQL)
  - not a general purpose database but a limited function database
  - usually distributed (client/server) with a defined API interface (LDAP)
  - security based on authentication (network security) and ACLs (access control lists)

IBM Technical Support

- A directory is nothing more than a database. LDAP is the standardized interface (X.500) to the LDAP directory. LDAP is not the directory but the defined APIs to gain access to the data within the directory (or database). The directory is also referred to as the backend store. To mean LDAP standards anything can be the backend store - the directory - as long as the directory can handle the LDAP interfaces. For the IBM solution to LDAP, DB2 V5+ and/or RACF is used as the backend store. Both of these support the IBM LDAP interfaces.
- de-facto Internet (TCP/IP-based) wire protocol for accessing and updating directory information
- "V2" defined in Internet Drafts
- "V3" defined in IETF RFCs 2251-2256

Traditional LDAP Environment

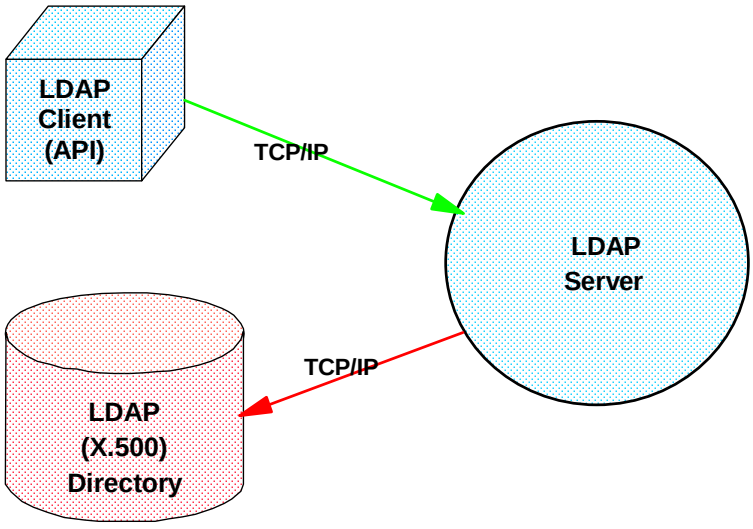


IBM Technical Support

- DAP (Directory Access Protocol) leads to LDAP
- History & Standards
- CCITT & OSI => formal standards - developed x.500 in 1988 - ISO 9594 in 1990 - usually very slow to develop and covered all possibilities
- The OSI (X.500) held the communication capabilities for all protocols that were possible. Therefore it was very 'heavy' and required lots of resources. It had its own communication stack (that is, it did not use the TCP/IP stack).
- DARPA & IAB & IETF => standards for the Internet, therefore they need to be quickly developed (Internet Drafts => RFCs => STDs) - RFC 1777, 1778, 1779, 1959, 1960 (LDAP V2 - status of draft standard) - RFC 2251 = LDAP V3 - status of proposed standard)
- Differences between V2 & V3

✓ Referrals (refer client to another server)
✓ Security (data protection - SASL)
✓ Internationalization (unicode support)
✓ Extensibility (dynamically defined objects)

Stand-Alone LDAP Environment



IBM Technical Support

- The DAP Server communication stack was too large to run on most system so the industry standards removed some protocols that were not used. As the DAP stack was reduced the new industry standard stack was referred to as LDAP (Lightweight DAP). The new LDAP stack was very similar to the standard TCP/IP stack, therefore most implementations of LDAP use what is referred to as a Stand-Alone LDAP Environment with the TCP/IP stack. Thus the LDAP Server and the X.500 Server have been combined. This is a picture of the most common LDAP industry solutions including the IBM solution.

Why is a Directory Service Important?

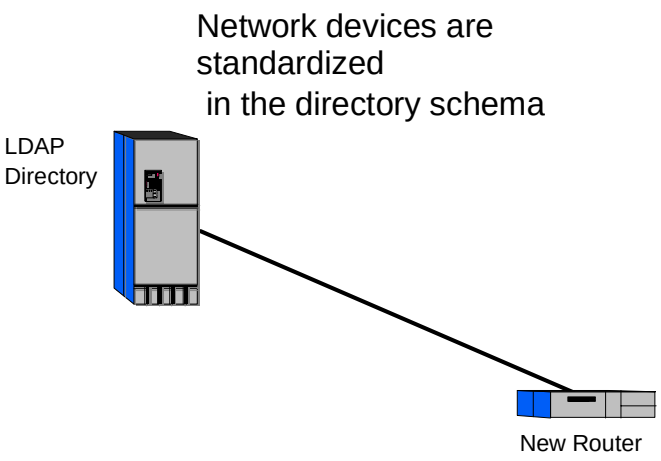
- **Example - Domain Name Service (DNS). We use it everyday - without it we wouldn't find services on the Internet.**
- **Within an Intranet or across the Internet there is a need to provide "locating information". Example - IBM BluePages.**
- **In addition, remote, distributed, single point of control is necessary for Enterprise Management. Example - DEN (Directory Enabled Network).**
- **Some view this as the key to PKI (Public Key Infrastructure) and Single Sign-On.**

IBM Technical Support

- Withour realizing it, we are using Directory Services, wether it is on the Internet or at work when we need to get some information.
- Although today most of these directories are seperate entities and not in any sense connected or related to one another. That is were the change is coming ....
- LDAP is viewed as a key in easing the management of many different components of our distributed systems. It may also provide the capabilities to central the management of these distributed systems without reducing the security or raising the complexity.

DEN Initiative !

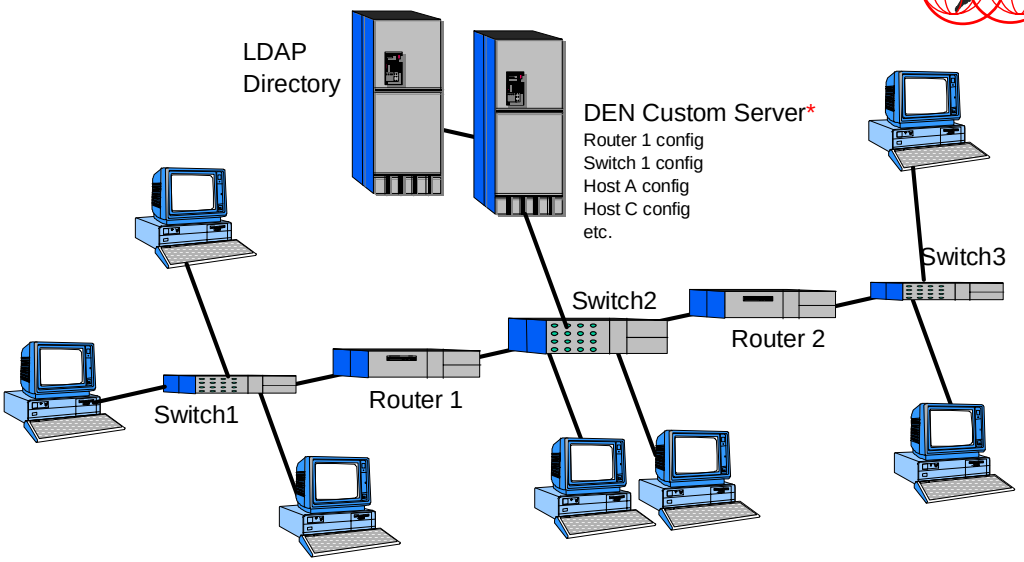
DEN Initiative



IBM Technical Support

- May 1997 - Microsoft formed the Directory Enabled network (DEN) Initiative - purpose "outline directory schema to include network devices"
- May 1997 - Cisco licenses Active Directory from Microsoft to compete against Netscape on Unix
- September 1997 - IBM joins Microsoft to develop Active Directory
- February 1998 - DEN moved under Desktop Management Task Force (DMTF)
- January 1999 - Cisco announces Active Directory on Unix
- February 1999- 3Com introduces first DEN product
- March 22, 1999 - IBM announces a complete family of DEN routers, switches and controllers.
- Lucent will introduce a switch in late 1999

DEN Custom Server Architecture

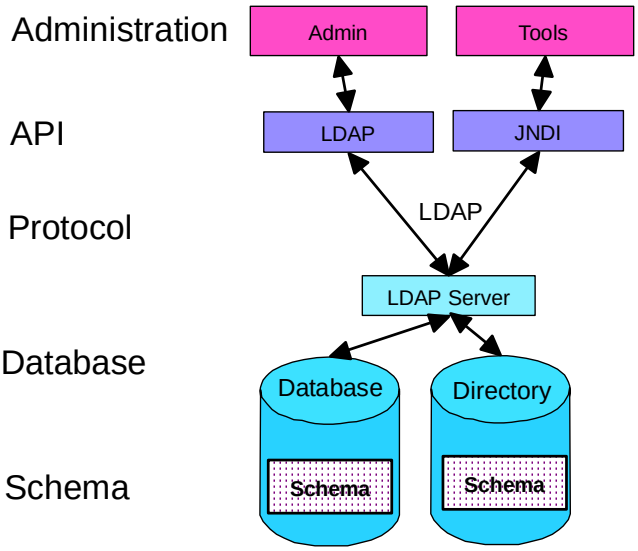


* **Patent Pending**

IBM Technical Support

LDAP Components

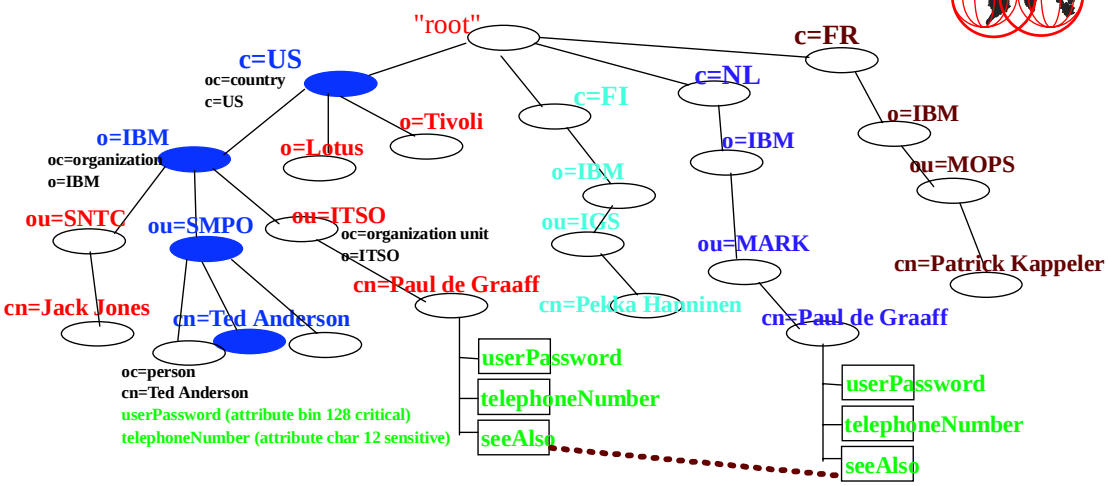
LDAP Components



IBM Technical Support

- This foil gives an overview of all the components and where they fit. The LDAP Protocol defines the LDAP standards. The LDAP Server provides these LDAP standard interface, that is provides program support (APIs) for the LDAP standards. Each LDAP solution will support its own backend store. In the case of the IBM solution, the backend store is either DB2 V5+ or RACF or both. The data in the directory must have a schema (that is, a data definition or layout) that means the LDAP standards. The last part component that is required for a LDAP solution is the API or the programming support. In the IBM solution, the supported languages are C, C++, and Java (that is, the Java Name Directory Interface). The administration tools will vary based upon the solution and the platform that is selected. Several vendors are building administration tools for their versions of LDAP: Novell Console1, IBM Concept X, Tivoli TME, etc.

LDAP Directory Content



- **DN: cn=Ted Anderson, ou=SMPO, o=IBM, c=US**
- **All entries have attributes (and values)**
- **Object class (oc) is an attribute in all entries**
- **Attributes grouped into mandatory and optional**
- **Attributes protected by Access Control Lists (ACLs)**

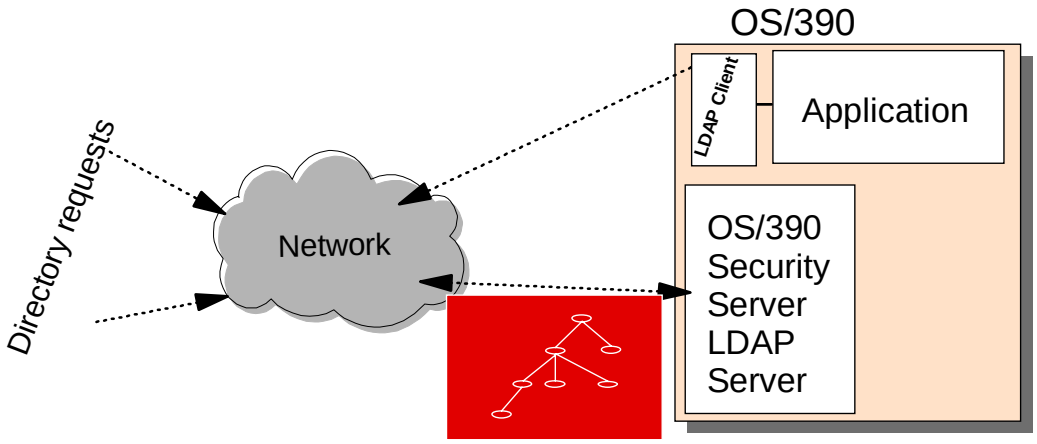
IBM Technical Support

- Heres a picture of an X.500 Directory model.
-
- Directory is a hierarchy of entries.
- Entries contain attributes.
- Attributes have one or more values.
- An entry's attributes (not their values) are defined by the entry's object class.
- Each entry has a name relative to its parent. This is a relative distinguished name (RDN).
- All RDNs from root to entry put together form a distinguished name (DN).
- RDN: cn=Tim Hahn
- DN: cn=Tim Hahn, o=IBM, c=US

The History of LDAP on OS/390

Brief History of LDAP on OS/390

OS/390 2.5 LDAP Support

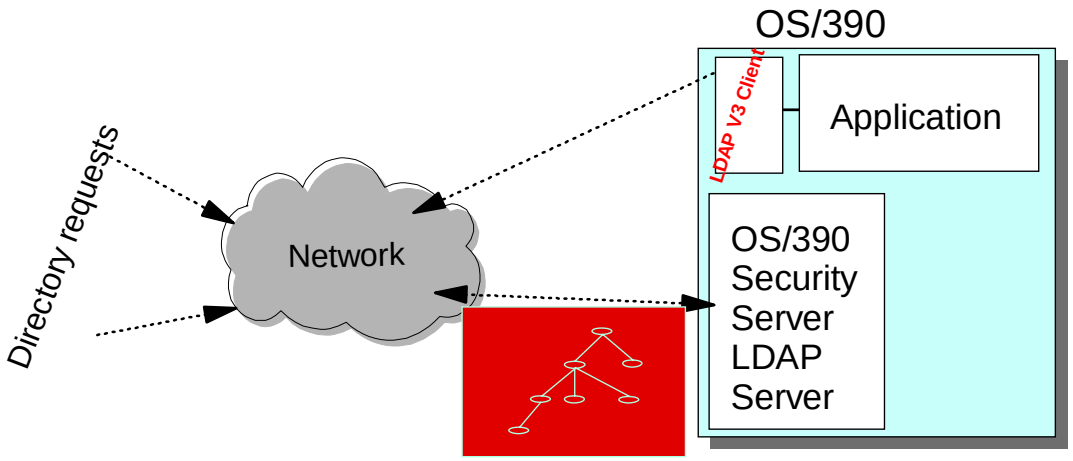


Complete LDAP V2 Support (Both Client & Server)
Directory Based on DB2 V5

IBM Technical Support

- In OS/390 2.4, LDAP V2 client support was introduced.
- In OS/390 2.5, which GA'ed in Sept 1998, LDAP V2 server support was introduced with the OS/390 Security Server. Both DB2 V5 and the OS/390 Security Server are required.
- Server includes DB2 backing store, access control and replication support
- OS/390 provides services for accessing directory information (LDAP client for C/C++) with OS/390 R4, GA'd 9/1997, V2 protocol and OS/390 R5, GA'd 3/1998

OS/390 2.6 LDAP Support

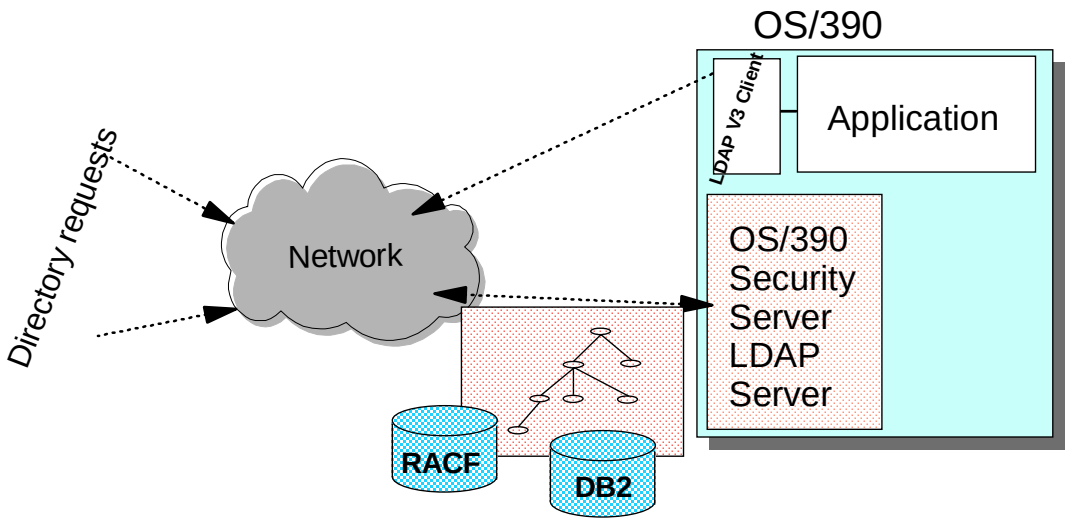


Added LDAP V3 Client Support

IBM Technical Support

- With OS/390 R6, which GA'd 6/1998, support was added for the LDAP V3 client with C/C++. The LDAP Server was updated to support advanced functions such as remote ACL Administration.

OS/390 2.7 LDAP Support

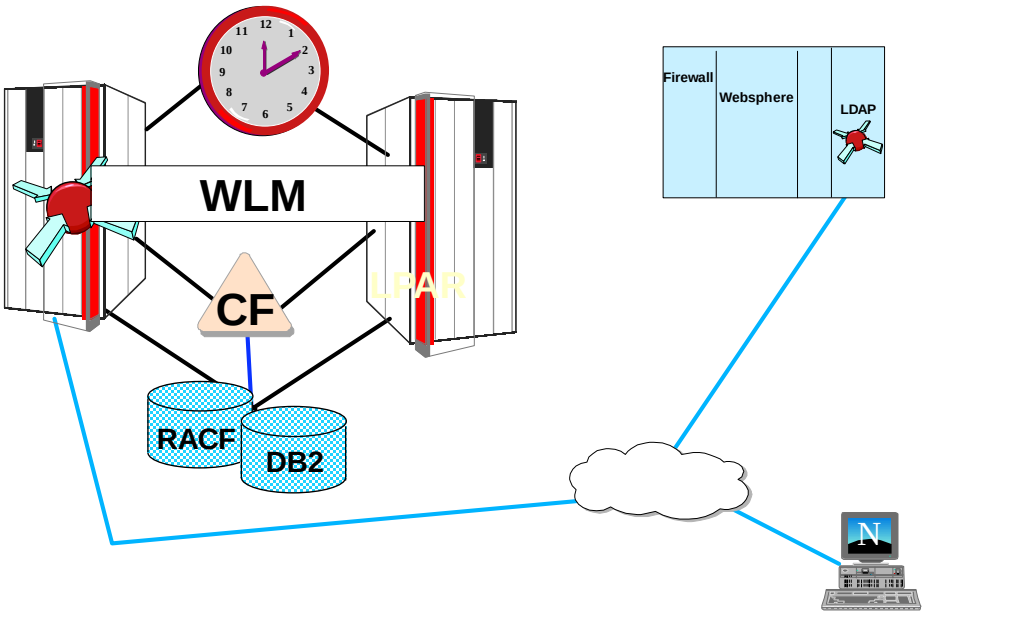


Added Sysplex Support to the LDAP V2 Server
Added Security Support to the LDAP V2 Server

IBM Technical Support

- With OS/390 R7, which GA'd 3/1999, the LDAP Server is still the LDAP V2 protocol but parallel sysplex support has been added. And the ability to use RACF as the backend store has been added. The parallel sysplex support means that the same DB2 or RACF database can be accessed by several different OS/390 LDAP Servers as their directories. This provides a higher grade of availability. Using RACF as the ldap directory means that, not only can a user (with the correct authority) access the RACF user and group information, but the LDAP Server can use RACF to authenicate (a LDAP bind) a user.
- OS/390 R7 provides services for accessing directory information from the LDAP client with C/C++ and has also added LDAP support for Java (JNDI) for both LDAP V2 & V3 protocols.

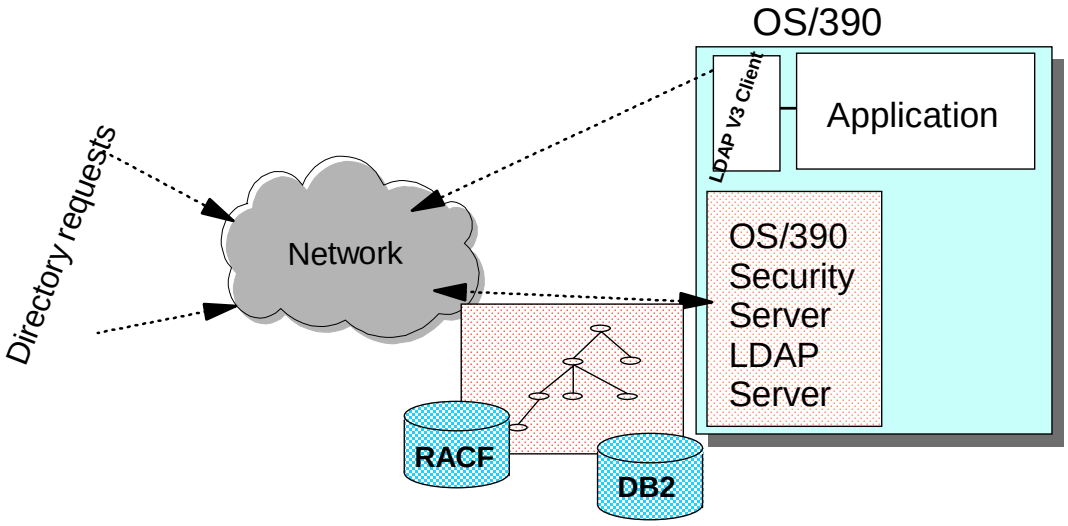
Parallel Sysplex Support in OS/390 R7



IBM Technical Support

- With the parallel sysplex support added in OS/390 R7, there is added availability. That is, several different LDAP Servers can use the same backend database (either RACF and/or DB2) with integrity. LDAP clients can make a query through one LDAP Server on a parallel sysplex and, if that LDAP Server is overloaded or down for maintenance, the result can be sent back through another LDAP Server within the parallel sysplex as long as the environment is set up correctly. Both RACF and DB2 support datasharing mode in a parallel sysplex.

OS/390 2.8 LDAP Support

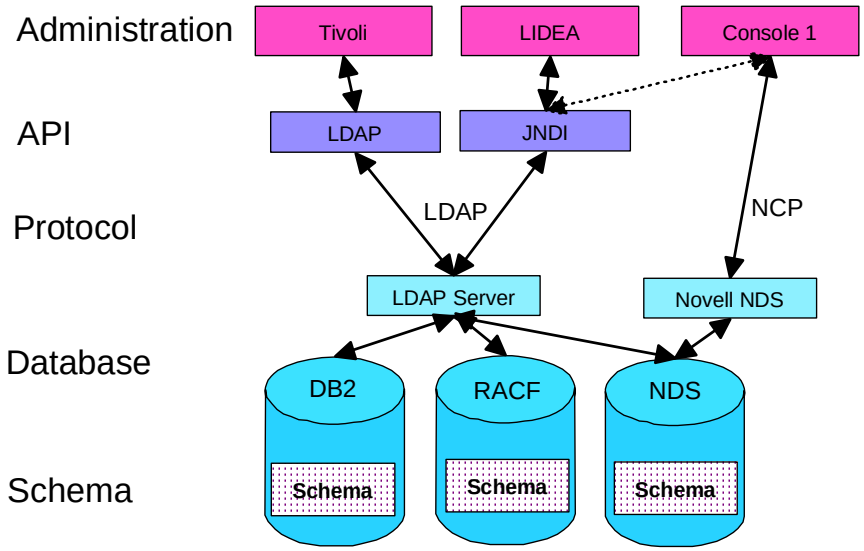


Complete the first phase of the LDAP V3 Server Support

IBM Technical Support

- LDAP V3 Protocol Support
- RootDSE and Controls support
- How to use rootDSE and Controls support
- Certificate Bind Support
- How to use Certificate Bind Support
- Internationalization/UTF-8 support
- How to use UTF-8 support
- Packaging changes for LDAP in V2R8

On OS/390 (Today and Tomorrow)



IBM Technical Support

- This foils gives an overview of all the components and where they fit. Starting with the administration piece and depending on the interface used, using a specific API to administer the database.
- Directory Service components on OS/390
- API
- C/C++: since OS/390 R4
- Java: JNDI in OS/390 R7
- Protocol
- LDAP
- Database
- LDAP Server: in Security Server, since OS/390 R5 (DB2)
- Novell NDS 4.1: GA 1Q99
- Schema
- Working to help define the eNetwork Schema
- Administration
- Working with Systems Management team to ensure Directory-enablement

LDAP V3 Protocol Support

- **Major elements of the LDAP V3 protocol include:**
  - **ability to obtain support information from server (rootDSE)**
  - **standardized referral support**
  - **operational controls**
  - **ability to bind using a certificate**
  - **data 'on-the-wire' in UTF-8 format**
- **V3 protocol is invoked in an application by setting the version referenced by the LDAP handle to version 3**

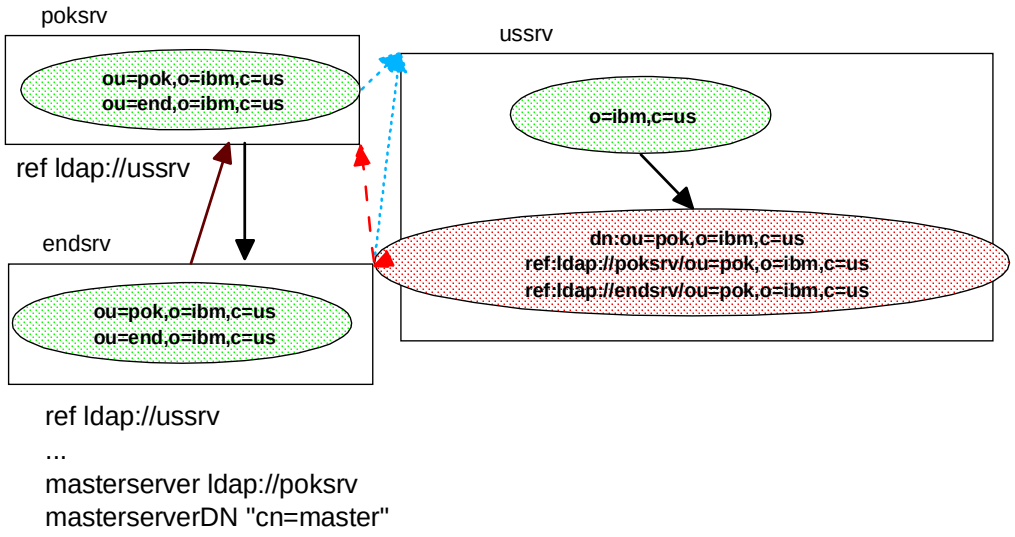
IBM Technical Support

- RootDSE Support
- A V3 Server will report support information, such as:
- LDAP versions supported
- Naming contexts managed
- Controls supported
- Applications can use to direct requests
- Information is retrieved by an LDAP base search request with search-base of the null string, e.g.
- ldapsearch -host V3host -Version 3 -scope base -base ""
- "objectclass=*"
- OS/390 rootDSE reports: versions, naming contexts, controls, referrals. Other items are defined in the RFC, but anything not supported by server is not to be returned.
- Must specify version 3 to obtain the rootDSE information.
-
- LDAP Referral Support
- Discussing referrals here to set the stage for discussion of controls.
- NOTE: In V2R8, there has been a separation of the specification of the location of the master server for a replica. Prior to R8, the default referral indicated the master server location, but this was really an 'overload' of the default referral. In R8, a new configuration file option has been added, 'masterserver', which takes an LDAP URL and indicates the location of the master server for a replica. If 'masterserver' is not specified, the default referral will still be used to allow compatability with the old method.
- Allows multiple servers to be used to manage different parts of the LDAP namespace
- Behavior standardized in LDAP V3 protocol Internet draft
- Default referral should point "up" to "parent" namespace server
- Referral objects should point "down" to "child" namespace server(s)

LDAP V2R8 Enhancements

Referral Support Example

Example using referrals and replication



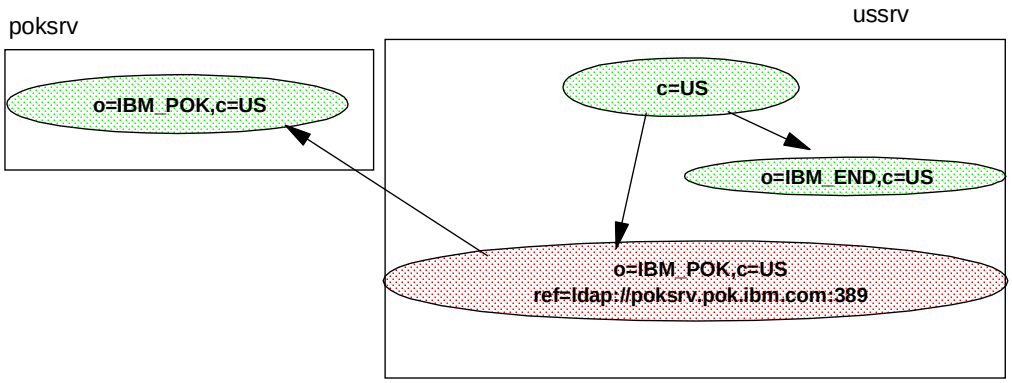
ref ldap://ussrv
...
masterserver ldap://poksrv
masterserverDN "cn=master"

IBM Technical Support

- Green dashed arrows indicate default referral. Red solid arrow indicates master server for replica. Pink dashed arrows indicate referral objects. Purple solid arrow indicates direction of replication.
- Updates sent to endsrv will be forwarded to poksrv, updates occur on poksrv and are replicated to endsrv.
- If a search is directed to poksrv or endsrv with ou=ral,o=ibm,c=us, referral will occur to ussrv. If ussrv has a referral object for ou=ral pointing to another server, that referral will also be followed. If ussrv has no referral object and does not itself manage ou=ral,o=ibm,c=us entities, 'no such object' is ultimately returned to the caller (search).
- If a search is directed to ussrv with ou=end,o=ibm,c=us, referral will occur to poksrv or endsrv. Choice is random. If one server is down the other will be contacted.

Using Controls Support

Example of using manageDSAIt control



To modify the IBM_POK referral object:

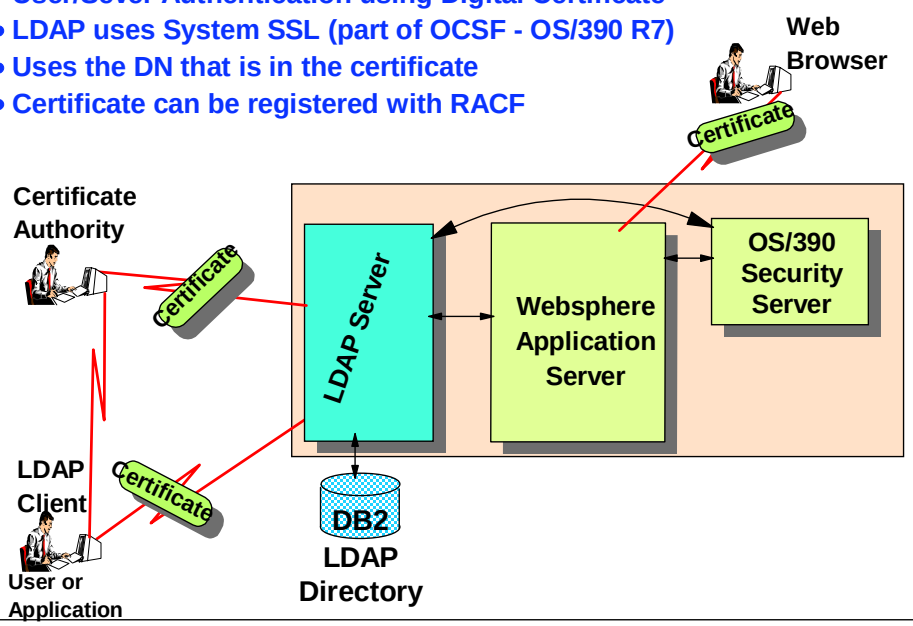
ldapmodify -h ussrv -V 3 -M -D <binddn> -w <bindpw> "o=IBM_POK,c=US" "ref=ldap://poksrv.pok.ibm.com:777"

IBM Technical Support

- Controls Support
- The only control currently defined by the IETF is the manageDSAIt control. Others being worked on include a sort control for search results. Companies can define their own controls. We have defined the authenticateOnly control to improve bind support.
- If the authenticateOnly control is present, no group membership information will be gathered. If it is not present, group membership information will be gathered based on the server configuration established by the administrator. For example, a server running with both an RDBM and SDBM backend and configured to gather groups from both, binding as the racf userid (SDBM bind) could gather group membership from both RDBM and SDBM(RACF) if the authenticateOnly control is absent. For an application that is only interested in using LDAP SDBM to authenticate using RACF userid/password, authenticateOnly can be used to limit bind processing to authentication and eliminate group membership gathering.
- Controls can be used to modify the behavior of an operation
- Few are formally defined today
- OS/390 LDAP Server supports these:
- manageDSAIt - causes referral object to be treated as an entry and not chased; allows update of referral objects via LDAP APIs
- authenticateOnly - suppresses gathering of group membership on bind

Certificate Bind Support

- User/Sever Authentication using Digital Certificate
- LDAP uses System SSL (part of OCSF - OS/390 R7)
- Uses the DN that is in the certificate
- Certificate can be registered with RACF

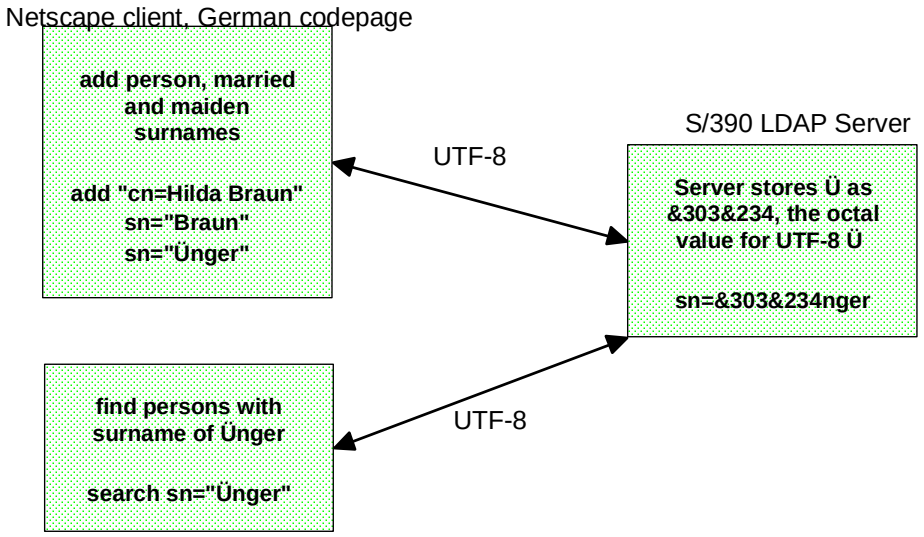


IBM Technical Support

- Allows applications to use certificates generated by a CA.
- Verifies both client and server are who they say they are. Uses SystemSSL functions(part of OCSF). LDAP Server must be configured to use certificates (SSL) and LDAP clients must understand and trust the certificates. Client application indicates use of a certificate on the bind operation by specifying bind method as 'external'. Bind DN taken from certificate. Verification that client and server are who they say they are is done during the SSL handshake, which occurs when the application sets up SSL communications using the SSL init API. What happens:
- SSL handshake occurs when the ssl init API is called
- Authentication occurs during the handshake and succeeds only if authentication succeeds
- Bind method is specified as "EXTERNAL" on the bind API call
- Certificate from handshake is used on bind
- Bind occurs using DN in the certificate
- IBM servers gather group membership information based on DN naming context

Using UTF-8 Support

Example of working with an entry containing UTF-8 data



IBM Technical Support

- For values stored in DB2's searchable tables, the server escapes 'non-portable' characters using an EBCDIC ampersand '&' character, followed by the 3 EBCDIC digits which represent the escaped character's octal value. Existing data consisting of '&nnn' where nnn is from '000' to '377' will need to be converted in order to achieve predictable search results. Documentation of the steps to take to determine if migration is needed, and how to migrate, will be provided in the LDAP Server book.
- Due to the data representation changes, including the above as well as the different 'wire' representations between V2 and V3 data, the database version for R8 will change. When ready to store UTF-8 data, database version must be migrated and other migration actions must be taken.
- Note that the sn in the searchable attribute table is stored as &303&234nger by the S390 LDAP Server.
- LDAP V3 Protocol requires data 'on-the-wire' to be in UTF-8 format
- UTF-8 allows representation of native language data
- Server management of UTF-8 data allows storage and retrieval of data regardless of language
- No special action needed to use support
- Migration actions may be needed

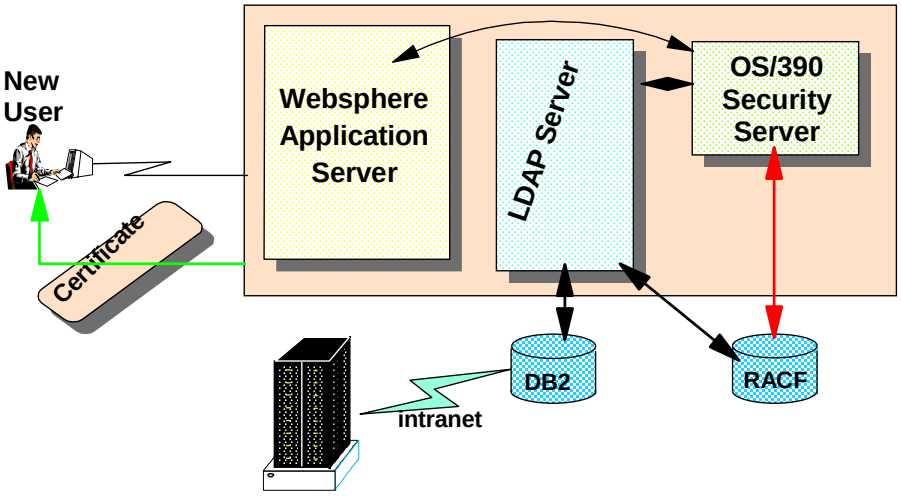
Packaging Changes for LDAP in V2R8

- **LDAP Client moves to OS/390 Security Server from OS/390 Base with DCE in same FMID as LDAP Server (HRSL180)**
  - **LDAP Client and Server ship as ALWAYS ENABLED(free of charge!)**
  - **For customers to use LDAP client or server, MUST install OS/390 Security Server**

- As in other releases of LDAP, the LDAP Client DLLs install into LPALIB. One of these DLL's in renamed in R8 from GLDMESG to GLDCMMN. The other remains GLDCLDAP.
- The LDAP Server does not install into LPALIB.
- One new DLL is shipped and installed with the LDAP Server, GLDBCDBM.
- All message catalogs have been renamed to include the product prefix "gld".
- Several additional, optional, configuration files containing schema information are shipped in HFS.

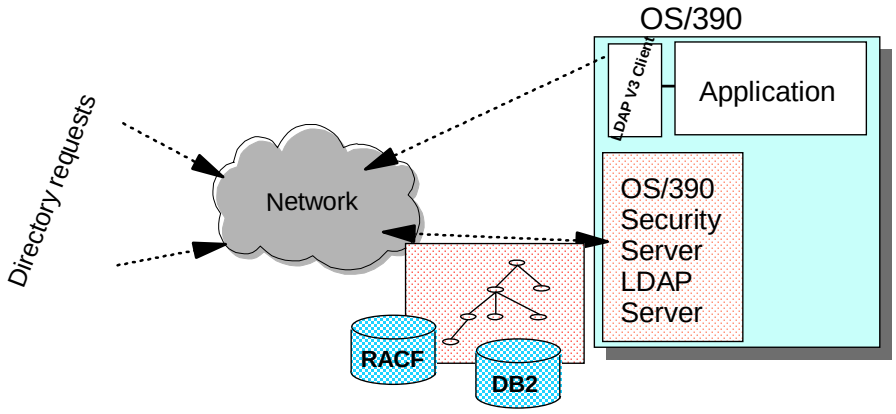
LDAP as a Security Tool

Possibility for User Administration?



LDAP as a Security Tool

- Authenication of Users (Single Sign-on?)
- Support for Digital Certificates
  - System SSL Support
- Controling Access to Information within the LDAP Server (ACLs)
- RACF (OS/390 only) Access and Information



The Next Step ...

- **Tactical ... V3 Schema Support in LDAP Server (OS/390)**

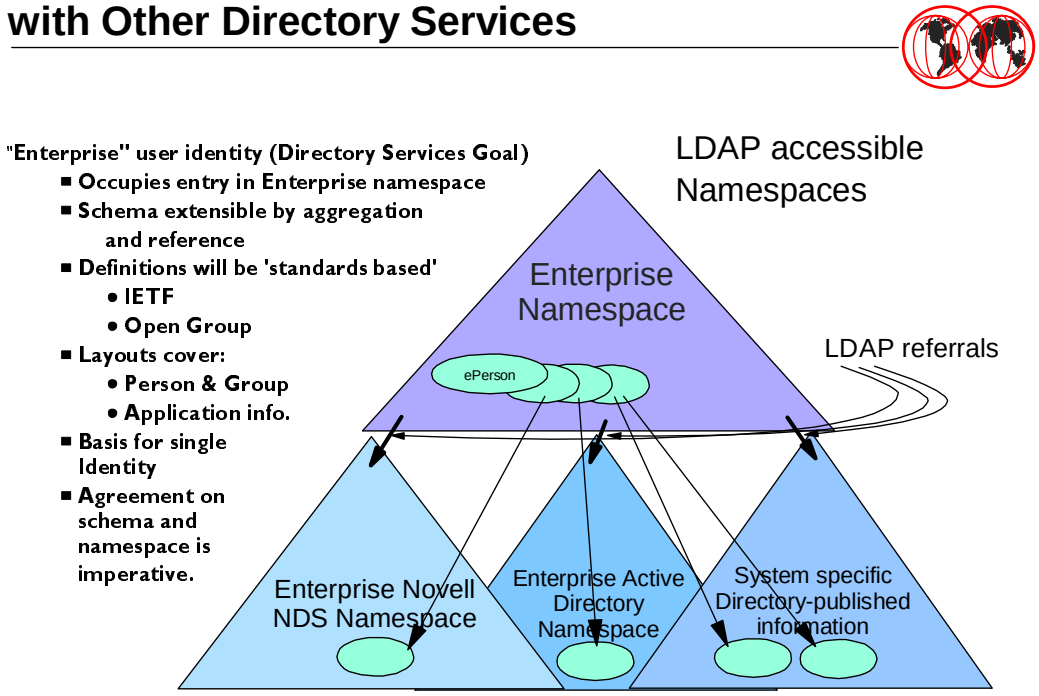
- **Strategically ...**
  - **Simplify Administration Across Directories**
  - **Raise the Level of Trust Between Systems**

IBM Technical Support

- Industry has rallies around something called a "meta-directory" - Really an extraction and sync utility for keeping multiple (separate before) directories in sync
- An example would be a program which keeps CERIS, CALLUP, and the IBMUS Name and Address book in sync
-
- A "meta-directory" is what our customers have been asking us for years. It would require vendor and industry agreement on one schema, although work is in progress it will take some time before this is finalized.

- The directory of directories will be more like a synchronizing tool to keep things in sync rather then the be all and end all directory !!

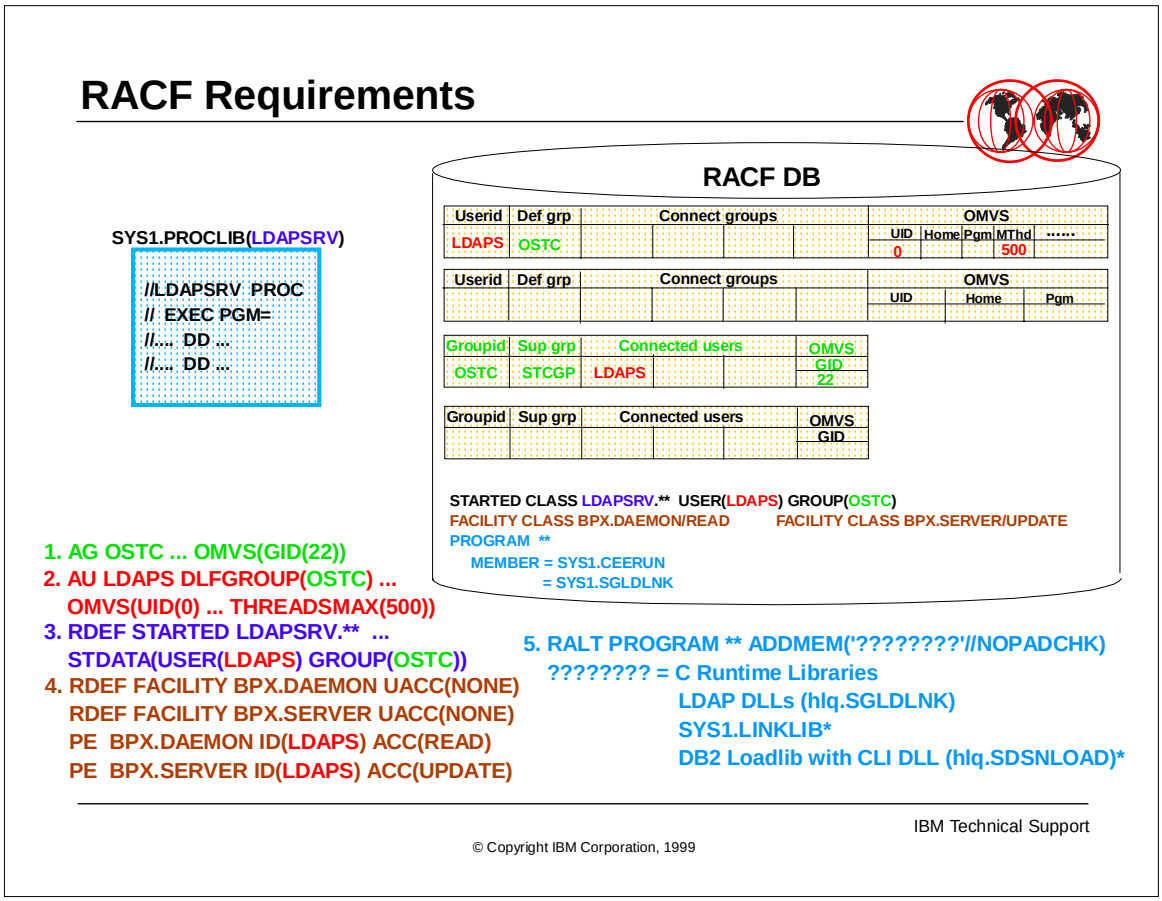
The Enterprise Namespace and Integration with Other Directory Services



IBM Technical Support

- An Enterprise namespace needs to cover more than just OS/390
- It must cover the computing resources of the entire enterprise, including existing workgroup servers
- Many Directory Services are "publishing" their information via the LDAP protocol:
  - IBM eNetwork LDAP Directory
  - IBM eNetwork X.500 Directory
  - Novell NDS
  - Microsoft Active Directory
  - Netscape Directory Server
  - Lotus Notes Name and Address Book

Implementation examples

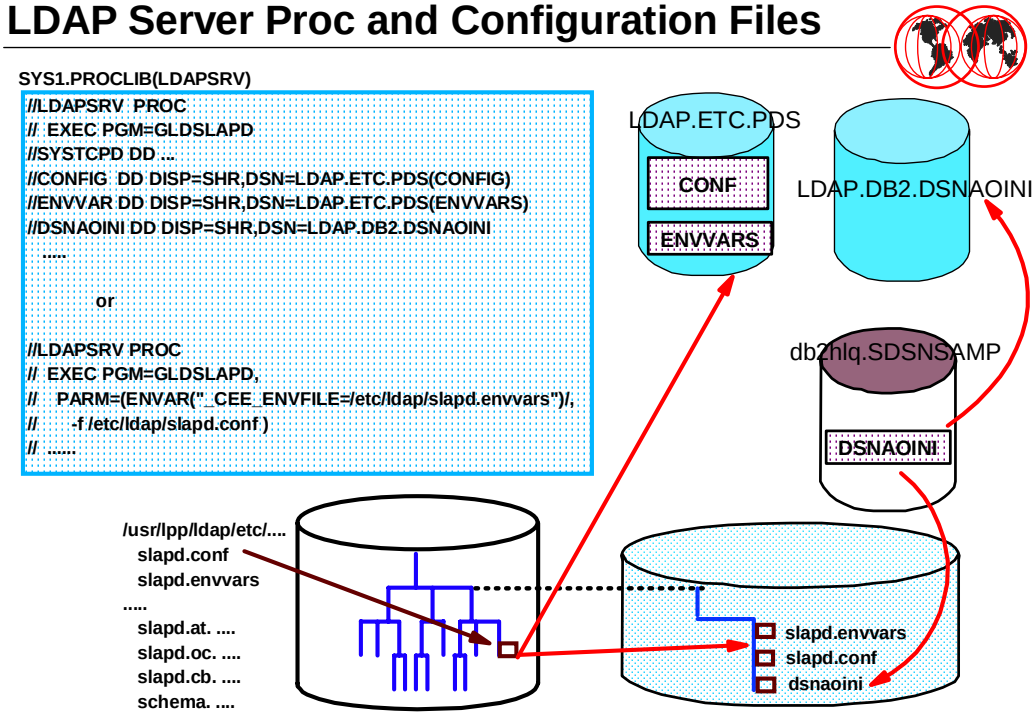


1. AU ldaps DEFLTGRP(ostc) NOTSO OMVS(UID(0) ...) ...
2. AG ostc SUPGROUP(stcgrp) ... OMVS(GID(22))
3. RDEF LDAPSRV.** CLASS(STARTED) STDATA(USER(ldaps) GROUP(ostc) ...) ...
4. PE BPX.DAEMON CLASS(FACILITY) ID(ldaps) ACC(READ)

➢PE BPX.SERVER CLASS(FACILITY) ID(ldaps) ACC(UPDATE)

The following RACF commands should have been issued before (they only have to be issued once):
RDEF BPX.DAEMON UACC(NONE) CLASS(FACILITY)
RDEF BPX.SERVER UACC(NONE) CLASS(FACILITY)
SETR CLASSACT(STARTED FACILITY)
SETR RACLIST(STARTED FACILITY)
The following RACF command needed to be issued after the FACILITY and STARTED classes are updated:
SETR RACLIST(STARTED FACILITY) REFRESH

LDAP Server Proc and Configuration Files



IBM Technical Support

- This foil gives an overview of all the components and where they fit. The LDAP Protocol defines the LDAP standards. The LDAP Server provides these LDAP standard interface, that is provides program support (APIs) for the LDAP standards. Each LDAP solution will support its own backend store. In the case of the IBM solution, the backend store is either DB2 V5+ or RACF or both. The data in the directory must have a schema (that is, a data definition or layout) that means the LDAP standards. The last part component that is required for a LDAP solution is the API or the programming support. In the IBM solution, the supported languages are C, C++, and Java (that is, the Java Name Directory Interface). The administration tools will vary based upon the solution and the platform that is selected. Several vendors are building administration tools for their versions of LDAP: Novell Console1, IBM Concept X, Tivoli TME, etc.

slapd.conf

slapd.conf (for global parms)

```
include                 /usr/lpp/ldap/etc/slapd.at.system
include                 /usr/lpp/ldap/etc/slapd.at.conf
include                 /usr/lpp/ldap/etc/slapd.oc.system
include                 /usr/lpp/ldap/etc/slapd.oc.conf
include                 /usr/lpp/ldap/etc/slapd.at.racf
include                 /usr/lpp/ldap/etc/slapd.oc.racf
port                    389
securePort              636
security                ssl
sslKeyRingFile          /etc/ldap/secure/ldapkeys.kyr
sslKeyRingFilePW        keypasswd
sslCipherSpecs          12288
maxthreads              500
maxconnections          1000
waitingthreads          500
timelimit               3600
sizelimit               500
adminDN     "cn=LDAP Admin,ou=ITSO,o=IBM,c=US"
adminPW                 admpasswd
```

slapd.conf (for RDBM)

```
database rdbm     GLDBRDBM
servername        ITSOLDAP
databasename      LDAPDB
dbuserid          LDAPADM
tbspaceentry      LDAPV27T
tbspace32k        LDAP32K
tbspace4k         LDAP4K
dsnaoini          LDAP.DB2.DSNAOINI
suffix            "cn=localhost"
suffix            "o=IBM,c=US"
index  cn   eq,sub
index  ou   eq,sub
index  sn   eq,sub
index  telephoneNumber  eq,sub
index  title  eq,sub
readOnly off
```

slapd.conf (for SDBM)

```
database sdbm   GLDBSDBM
suffix          "sysplex=LOCAL,c=US"
```

IBM Technical Support

- This foil gives an overview of all the components and where they fit. The LDAP Protocol defines the LDAP standards. The LDAP Server provides these LDAP standard interface, that is provides program support (APIs) for the LDAP standards. Each LDAP solution will support its own backend store. In the case of the IBM solution, the backend store is either DB2 V5+ or RACF or both. The data in the directory must have a schema (that is, a data definition or layout) that means the LDAP standards. The last part component that is required for a LDAP solution is the API or the programming support. In the IBM solution, the supported languages are C, C++, and Java (that is, the Java Name Directory Interface). The administration tools will vary based upon the solution and the platform that is selected. Several vendors are building administration tools for their versions of LDAP: Novell Console1, IBM Concept X, Tivoli TME, etc.

slapd.envvars and dsnaoini

dsnaoini

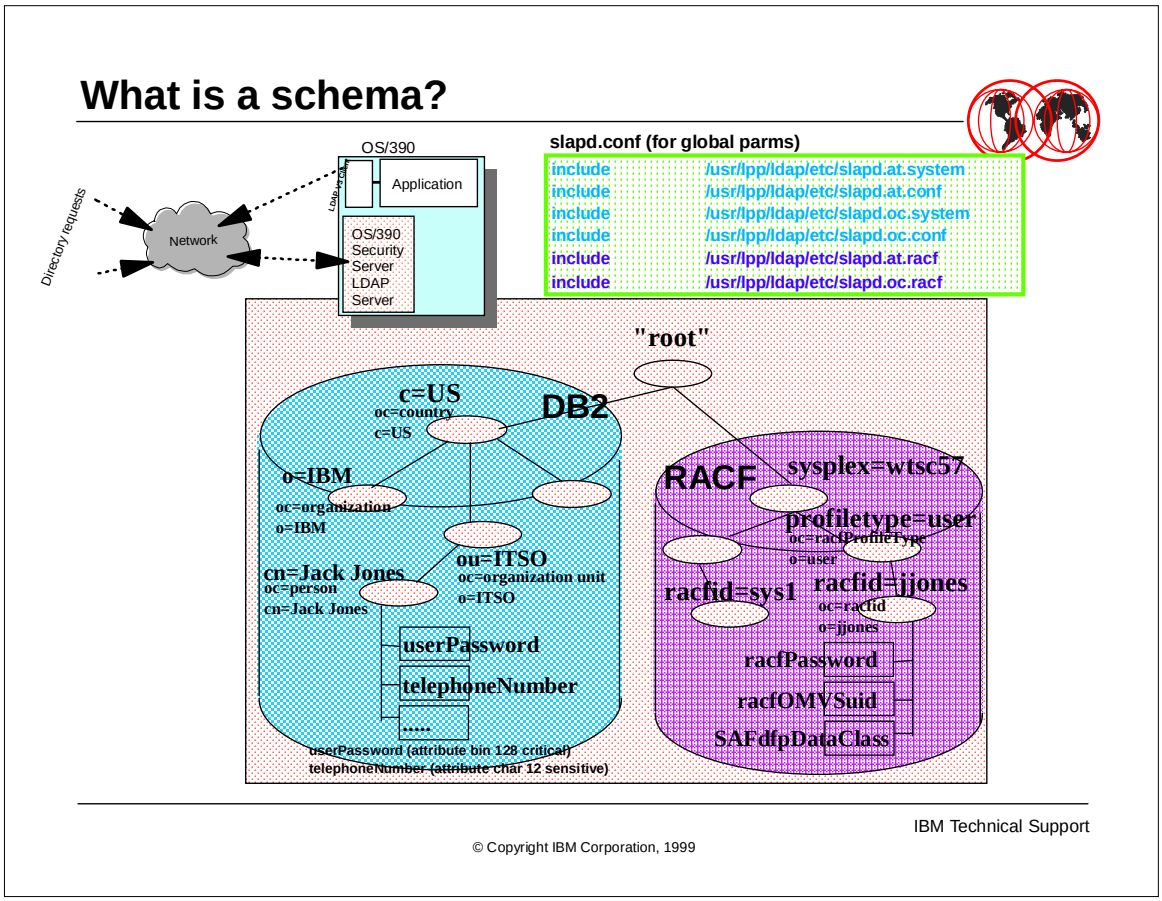
```
; This is a comment line...
; Example COMMON stanza
ÝCOMMON¨
MVSDEFAULTSSID=V51A

; Example SUBSYSTEM stanza for V42A subsystem
ÝV51A¨
MVSATTACHTYPE=CAF
PLANNAME=DSNACLI

; Example DATA SOURCE stanza for STLEC1 data source
ÝSTLEC1¨
AUTOCOMMIT=0
CONNECTTYPE=2

; Example DATA SOURCE stanza for STLEC1B data source
ÝSTLEC1B¨
CONNECTTYPE=2
CURSORHOLD=0
```

slapd.envvars

```
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/En_US.IBM-1047/%N
LANG=En_US.IBM-1047
```

IBM Technical Support

- This foil gives an overview of all the components and where they fit. The LDAP Protocol defines the LDAP standards. The LDAP Server provides these LDAP standard interface, that is provides program support (APIs) for the LDAP standards. Each LDAP solution will support its own backend store. In the case of the IBM solution, the backend store is either DB2 V5+ or RACF or both. The data in the directory must have a schema (that is, a data definition or layout) that means the LDAP standards. The last part component that is required for a LDAP solution is the API or the programming support. In the IBM solution, the supported languages are C, C++, and Java (that is, the Java Name Directory Interface). The administration tools will vary based upon the solution and the platform that is selected. Several vendors are building administration tools for their versions of LDAP: Novell Console1, IBM Concept X, Tivoli TME, etc.



With OS/390 R7, which GA'd 3/1999, the LDAP Server is still the LDAP V2 protocol but parallel sysplex support has been added. And the ability to use RACF as the backend store has been added. The parallel sysplex support means that the same DB2 or RACF database can be accessed by several different OS/390 LDAP Servers as their directories. This provides a higher grade of availability. Using RACF as the ldap directory means that, not only can a user (with the correct authority) access the RACF user and group information, but the LDAP Server can use RACF to authenticate (a LDAP bind) a user.
OS/390 R7 provides services for accessing directory information from the LDAP client with C/C++ and has also added LDAP support for Java (JNDI) for both LDAP V2 & V3 protocols.

slapd.oc.?????

slapd.oc.conf

```
objectclass top
  requires
     objectClass

objectclass country
  requires
     objectClass,
     c
  allows
     searchGuide,
     description

objectclass organization
  requires
     objectClass,
     o
  allows
     ....
     postalCode,
     street,
     telephoneNumber,
     .....
```

```
objectclass orgainzationalUnit
  requires
     objectClass,
     ou
  allows
     ....
     description,
     postalAddress,
     seeAlso,
     .....

objectclass person
  requires
     objectClass,
     sn,
     cn
  allows
     description,
     seeAlso,
     telephoneNumber,
     userPassword
```

- ✓ **objectclass definitions**
- ✓ **describes order and requirements**
- ✓ **same as schema.???.oc**
- ✓ **can be modified or user defined (but not in this file)**
- ✓ **also:**
  - **slapd.oc.system**
  - **slapd.cb.oc.conf**
  - **slapd.oc.racf**
- ✓ **or**
  - **schema.IBM.oc**
  - **schema.system.oc**
  - **schema.user.oc**

IBM Technical Support

- This foil gives an overview of all the components and where they fit. The LDAP Protocol defines the LDAP standards. The LDAP Server provides these LDAP standard interface, that is provides program support (APIs) for the LDAP standards. Each LDAP solution will support its own backend store. In the case of the IBM solution, the backend store is either DB2 V5+ or RACF or both. The data in the directory must have a schema (that is, a data definition or layout) that means the LDAP standards. The last part component that is required for a LDAP solution is the API or the programming support. In the IBM solution, the supported languages are C, C++, and Java (that is, the Java Name Directory Interface). The administration tools will vary based upon the solution and the platform that is selected. Several vendors are building administration tools for their versions of LDAP: Novell Console1, IBM Concept X, Tivoli TME, etc.

slapd.at.?????

slapd.at.conf

```
attribute objectClass                cis   objectclass  128  normal
attribute ......
attribute cn   commonName            cis   cn           128  normal
attribute sn   surName               cis   sn           128  normal
attribute o    organizationName      cis   o            128  normal
attribute description                cis   description  128  normal
attribute dn   distinguishedName     dn    dn           1000 normal
attribute userPassword               bin   userPassword 128  critical
attribute homePhone                  tel   homePhone    32   senitive
.....................
```

slapd.at.racf

```
attribute sysplex                    cis   _nocreate    8    senitive
attribute profileType                cis   _nocreate    5    senitive
attribute racfid                     cis   _nocreate    8    senitive
attribute racfInstallationData       cis   _nocreate    255  senitive
attribute racfPassword               cis   _nocreate    8    critical
attribute racfOMVSuid                cis   _nocreate    10   senitive
attribute SAFdfpStorageClass         cis   _nocreate    8    senitive
.....................
```

- ✓ attribute definitions
- ✓ all entries have attributes
- ✓ security controls set by attributes
- ✓ can be modified or user defined (but not in this file)
- ✓ also:
  - slapd.at.system
  - slapd.cb.at.conf
  - slapd.at.racf
- ✓ or
  - schema.IBM.at
  - schema.system.at
  - schema.user.at

- This foil gives an overview of all the components and where they fit. The LDAP Protocol defines the LDAP standards. The LDAP Server provides these LDAP standard interface, that is provides program support (APIs) for the LDAP standards. Each LDAP solution will support its own backend store. In the case of the IBM solution, the backend store is either DB2 V5+ or RACF or both. The data in the directory must have a schema (that is, a data definition or layout) that means the LDAP standards. The last part component that is required for a LDAP solution is the API or the programming support. In the IBM solution, the supported languages are C, C++, and Java (that is, the Java Name Directory Interface). The administration tools will vary based upon the solution and the platform that is selected. Several vendors are building administration tools for their versions of LDAP: Novell Console1, IBM Concept X, Tivoli TME, etc.

Running with SDBM only

slapd.conf (for SDBM)

```
include          /etc/ldap/slapd.at.racf
include          /etc/ldap/slapd.oc.racf

port             389
.................
security         ssl
.................
adminDN   "racfid=jjones,profiletype=user,sysplex=local"

database         sdbm GLDBSDBM
suffix           "sysplex=LOCAL"
```

- ✓RACF is only backend store
- ✓No DB2 set up required
- ✓Still need adminDN
- ✓Still need suffix
- ✓Still want SSL
- ✓Do not use adminPW
- ✓Identify that SDBM is being used

'SYS1.PROCLIB(LDAPSRV)'

```
//LDAPSRV  PROC REGSIZE=90M,OUTCLASS='S',DEBUG='-d 65519 -p 389'
//GO       EXEC PGM=GLDSLAPD,REGION=&REGSIZE,TIME=1440,
//         PARM=('&DEBUG')
//CONFIG   DD DSN=JJONES.LDAP.ETCPDS(STDCONF),DISP=SHR
//ENVVAR   DD DSN=JJONES.LDAP.ETCPDS(STDENV),DISP=SHR
//SLAPDOUT DD SYSOUT=&OUTCLASS
//SYSOUT   DD SYSOUT=&OUTCLASS
//SYSUDUMP DD SYSOUT=&OUTCLASS
//CEEDUMP  DD SYSOUT=&OUTCLASS
//SYSTCPD  DD DSN=TCPIP.INTRA.TCPPARMS(TCPDATA),DISP=SHR
```

IBM Technical Support

Be forewarned - you can front-end RACF with LDAP but this will not give you the same pinpoint control of the RACF environment that you know and love. Although the LDAP Server running with SDBM is very useful, it is recommended that the LDAP Server not be viewed as an administration tool for RACF. Also there are several fields and areas that will not be viewable from the LDAP environment. That said, there are several reasons why the LDAP Server might want to be used with RACF. To set up the LDAP Server with SDBM (RACF) follow the steps below:

1.

Running with RDBM only

CONF28

```
include          /etc/ldap/slapd.at.system
include          /etc/ldap/slapd.at.conf
include          /etc/ldap/slapd.oc.system
include          /etc/ldap/slapd.oc.conf

port             389
.................
security         ssl
.................
adminDN    "cn=LDAP Admin,ou=ITSO,o=IBM,c=US"
adminPW          admpasswd

database         rdbm GLDBRDBM
servername       CENTDB2
databasename     LDAP28
dbuserid         JJONES
tbspaceentry     LDAPTENT
tbspace32k       BIGTBLSP
tbspace4k        SMLTBLSP
tbspacemutex     MUTEXTBL
suffix           "cn=localhost"
suffix           "o=IBM_US,c=US"
index            cn  eq,sub
index            ou  eq,sub
index            sn  eq,sub
readOnly         off
```

- ✓ PROC needs to point to conf, envvar, and DB2 ini
- ✓ DB2 ini ==> subsystem & location
- ✓ conf ==> indicates DB2, location, database within subsystem, tablespaces within database, & userid for tables - information about LDAP Server set up

'SYS1.PROCLIB(LDAPSRV)'

```
//LDAPSRV  PROC ...
//CONF     DD  DSN=LDAP.ETCPDS(CONF28),....
//DNSAOINI DD  DSN=LDAP.SERVER.DNSAOINI,...
```

LDAP.SERVER.DSNAOINI

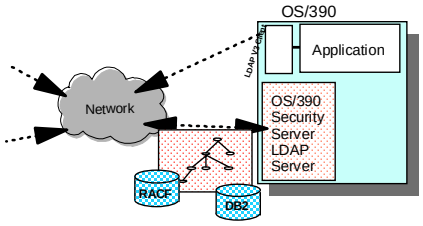
```
ÝCOMMON¨
MVSDEFAULTSSID=DB51

ÝDB51¨
MVSATTACHTYPE=CAF
PLANNAME=DSNACLI

ÝCENTDB2¨
AUTOCOMMIT=0
CONNECTTYPE=1
```


- This foil gives an overview of all the components and where they fit. The LDAP Protocol defines the LDAP standards. The LDAP Server provides these LDAP standard interface, that is provides program support (APIs) for the LDAP standards. Each LDAP solution will support its own backend store. In the case of the IBM solution, the backend store is either DB2 V5+ or RACF or both. The data in the directory must have a schema (that is, a data definition or layout) that means the LDAP standards. The last part component that is required for a LDAP solution is the API or the programming support. In the IBM solution, the supported languages are C, C++, and Java (that is, the Java Name Directory Interface). The administration tools will vary based upon the solution and the platform that is selected. Several vendors are building administration tools for their versions of LDAP: Novell Console1, IBM Concept X, Tivoli TME, etc.

Running with Both RDBM and SDBM

- **Combine the previous configuration files**
- **Use all the 'include' statements**
- **Use adminDN as RACF id with no adminPW**
- **Use one 'database sdbm' statement for RACF and one (or more) 'database rdbm' statement for each DB2 database**

```
include          /etc/ldap/slapd.at.system
include          /etc/ldap/slapd.at.racf
.....
include          /etc/ldap/slapd.oc.system
include          /etc/ldap/slapd.oc.racf

.................
adminDN    "racfid=jjones,profiletype=user,sysplex=local,c=US"

database         rdbm GLDBRDBM
servername       CENTDB2
......
database         sdbm GLDBSDBM
suffix           "sysplex=local,c=US"
......
```



Running with SSL

slapd.conf (for global parms)

```
include                 .....
include                 /etc/ldap/slapd.at.racf
include                 /etc/ldap/slapd.oc.racf
include                 .....

port                    389
secureport              636
security                ssl
sslAuth                 serverAuth
sslkeyringfile          /etc/ldap/secure/LdapRacfServer.kdb
#  sslkeyringfilePW  ????
sslkeyringPWstashfile  /etc/ldap/secure/LdapRacfServer.sth
sslcipherspecs          12288
adminDN    "racfid=jjones,profiletype=user,sysplex=local"


database
suffix
```

- ✓ **Which secure port - client needs to know**
- ✓ **How much security**
- ✓ **Server identification in a certificate**
- ✓ **Secure the server key database password**
- ✓ **What level of encryption will be supported**
- ✓ **Set sslAuth to serverClientAuth to allow clients to SASL bind to LDAP Server**
- ✓ **replKeyRingFile and repliKeyRingPW no longer supported**

IBM Technical Support

- This foil gives an overview of all the components and where they fit. The LDAP Protocol defines the LDAP standards. The LDAP Server provides these LDAP standard interface, that is provides program support (APIs) for the LDAP standards. Each LDAP solution will support its own backend store. In the case of the IBM solution, the backend store is either DB2 V5+ or RACF or both. The data in the directory must have a schema (that is, a data definition or layout) that means the LDAP standards. The last part component that is required for a LDAP solution is the API or the programming support. In the IBM solution, the supported languages are C, C++, and Java (that is, the Java Name Directory Interface). The administration tools will vary based upon the solution and the platform that is selected. Several vendors are building administration tools for their versions of LDAP: Novell Console1, IBM Concept X, Tivoli TME, etc.

LDAP command examples

Quick ldapsearch

- From UNIX System Services

ldapsearch -h wtsc57.itso.ibm.com
-D "racfid=JJONES,profiletype=user,sysplex=LOCAL" -w jackpwd
-b "ou=AUSTIN, o=IBM_US, c=US"
"objectclass=*"

- From TSO
  - Add xxx.SGLDEXEC to SYSEXEC in logon proc

===> LDAPSRCH -h wtsc57.itso.ibm.com
-D "racfid=jjones,profiletype=user,sysplex=local"
-w jackpwd -b "sysplex=local" "objectclass=*" dn

ldapsearch Output

From the TSO command:

```
.........
racfid=BPX,profiletype=USER,sysplex=LOCAL
racfid=BPXROOT,profiletype=USER,sysplex=LOCAL
.........
racfid=GIM,profiletype=GROUP,sysplex=LOCAL
racfid=GLD,profiletype=GROUP,sysplex=LOCAL
.........
sysplex=LOCAL
profiletype=user,sysplex=LOCAL
profiletype=group,sysplex=LOCAL
```

From the UNIX command:

```
.......
cn=Jack Jones, ou=Widget Division, ou=Austin, o=IBM_US, c=US
objectclass=newPilotPerson
cn=Jack Jones
sn=Jones
description=This is Jack the Ripper
homephone=293-1439
homepostaladdress=Poughkeepsie, New York
personaltitle=The Expert
mail=jjones@vnet.ibm.com

cn=httpserver diet,ou=AUSTIN,o=IBM_US,c=US
objectclass=organizationalperson
cn=httpserver diet
sn=httpserver diet
userpassword=test
```

ldapmodify Command

The UNIX command:

```
ldapmodify -h wtsc57.itso.ibm.com
-D  "cn=LDAP Admin,ou=ITSO,o=IBM,c=US" -w
adminpwd
-f /u/jjones/rdbmmod.mods
```

```
dn: cn=Jack Jones,ou=Widget Division,ou=Austin,o=IBM_US,c=US
changetype: modify
replace: userpassword
userpassword: racfgood
-
replace: mail
mail: johnjone@us.ibm.com
-
replace: description
description: Long live IBM's new retirement plan - it is the best.
-
```

Output from the UNIX command:

```
cn=Jack Jones, ou=Widget Division, ou=Austin, o=IBM_US, c=US
objectclass=newPilotPerson
cn=Jack Jones
sn=Jones
homephone=293-1439
homepostaladdress=Poughkeepsie, New York
personaltitle=The Expert
userpassword=racfgood
mail=johnjone@us.ibm.com
description=Long live IBM's new retirement plan - it is the best.
```

ldapadd Command

The UNIX command:

```
ldapadd    -h wtsc57.itso.ibm.com -Z -p 636
-D  "racfid=jjones,profiletype=user,sysplex=local" -w jackpwd
-K /u/jjones/secure/LdapClient.kdb -P racf
-f /u/jjones/racfadd.mods
```

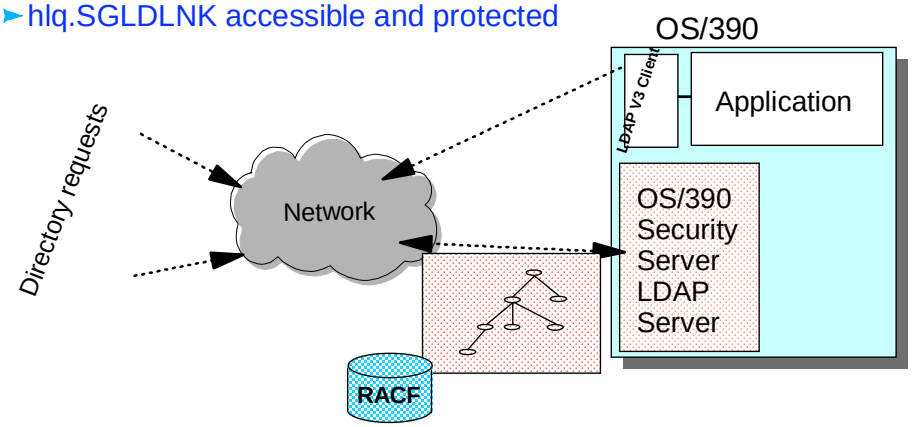
```
dn: racfid=thieflou,profiletype=user,sysplex=LOCAL
objectclass: racfUser
racfid: thieflou
racfPassword: secret
```

Output from the LU THIEFLOU command:

```
 USER=THIEFLOU  NAME=UNKNOWN  OWNER=JJONES    CREATED=99.186
  DEFAULT-GROUP=SYS1      PASSDATE=00.000  PASS-INTERVAL=180
  ATTRIBUTES=NONE
  REVOKE DATE=NONE   RESUME DATE=NONE
  LAST-ACCESS=UNKNOWN
  CLASS AUTHORIZATIONS=NONE
  NO-INSTALLATION-DATA
  NO-MODEL-NAME
  LOGON ALLOWED   (DAYS)        (TIME)
  ---------------------------------------------
  ANYDAY                        ANYTIME
   GROUP=SYS1      AUTH=USE      CONNECT-OWNER=JJONES
CONNECT-DATE=99.186
     CONNECTS=    00  UACC=NONE     LAST-CONNECT=UNKNOWN
     CONNECT ATTRIBUTES=NONE
     REVOKE DATE=NONE   RESUME DATE=NONE
  SECURITY-LEVEL=NONE SPECIFIED
  CATEGORY-AUTHORIZATION
   NONE SPECIFIED
  SECURITY-LABEL=NONE SPECIFIED
```

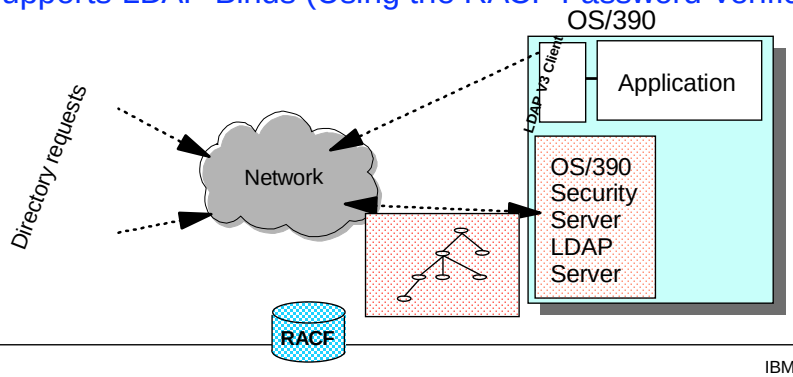
Using RACF as the SDBM

- Accessing RACF Information from the OS/390 LDAP Server
- Customization of the LDAP Server Considerations
  - Configuration files
  - slapd.at.racf, slapd.oc.racf, and slapd.conf
  - schema.IBM.at and schema.IBM.oc
  - hlq.SGLDLNK accessible and protected



RACF Functions that LDAP Server Supports

- User and Group Commands and Information
- Add or Delete Users and/or Groups
  - ADDUSER (AU) and DELUSER (DU) Commands
  - ADDGROUP (AG) and DELGROUP (DG) Commands
- Modify and Retrieve Information on Users and/or Groups
  - LISTUSER (LU) and ALTUSER (ALU) Commands
  - LISTGRP (LG) and ALTGROUP (ALG) Commands
- Supports LDAP Binds (Using the RACF Password Verification)



RACF Examples Using LDAP Commands

ldapmodify -h itso.ldap.srv -p 636 -D bindDN -w passwd -f mod.file -Z

```
dn: racfid=jjones,profiletype=user,sysplex=local
changetype: modify
racfOmvsHome: /u/jjones
racfBuilding: 707
SAFDefaultCommand: LOGOFF
```

Must have the RACF authority to issue the RACF command (in this case ALU).

RACF Examples Using LDAP Commands

ldapsearch -h itso.ldap.srv -p 636 -D bindDN -w passwd -Z
-b "racfid=jjones,profiletype=user,sysplex=local" "objectclass=*"

```
racfid=jjones,profiletype=USER,sysplex=local
objectclass=racfUser
...
racfid=jjones
racfauthorizationdate=99.134
racfdefaultgroup=racfid=GOODGUYS,profiletype=GROUP,sysplex=local
racfattributes =SPECIAL
racfrevokedate=NONE
safaccountnumber=75932
racfomvsuid=0
racfomvshome=/u/jjones
....
```

Operational LDAP considerations

Sample Server

/usr/lpp/ldap/examples/sample_server/
Select one or more files with / or act

Filename
_ dsnaoini.db2ini
_ dsntijcl.jcl
_ ldapspfi.spufi
_ README <=======
_ sample.ldif
_ slapd.at.conf
_ slapd.at.system
_ slapd.conf
_ slapd.oc.conf
_ slapd.oc.system

Quick and easy way to quickly implement a test LDAP Server. Documented in README file.

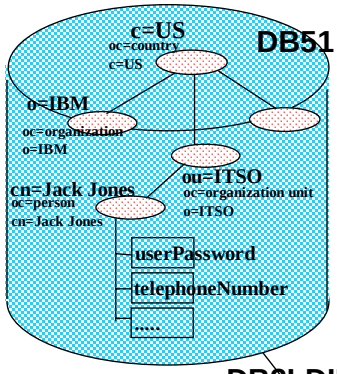
Starting and Stopping

- Decide if controlled with UNIX or OS/390
  - Define RACF userid for the LDAP Server
  - For OS/390:
    - Make executables accessible to the system (LNKLST)
    - Move sample PROC to SYS1.PROCLIB
    - Move the files to the production datasets or files
    - Configure the LDAP Server files
    - Issue the s ldapsrv with whatever parms are needed
    - Issue the p ldapsrv to stop the LDAP Server
    - Control with standard OS/390 tools
  - As a UNIX process:
    - Make executables accessible to the system (LNKLST)
    - Move the files to the production files
    - Configure the LDAP Server files
    - Logon as the LDAP Server user
    - Issue the /usr/sbin/sladp & with whatever parms are needed
    - Issue the grep -ef | grep slapd to get the process id (PID)
    - Issue the kill -15 *process-ID* to stop the LDAP Server

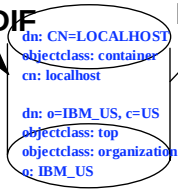
Migrating to new LDAP R8 schema

slapd.conf

```
include          /etc/ldap/slapd.at.system
include          /etc/ldap/slapd.oc.system
.....
```



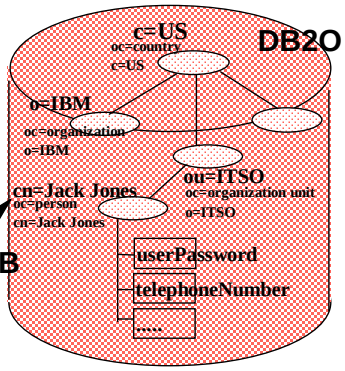
DB2LDIF



LDIF2DB

slapd.conf

```
include          /etc/ldap/schema.system.at
include          /etc/ldap/schema.system.oc
.....
```



*** Increase tablespace size.**

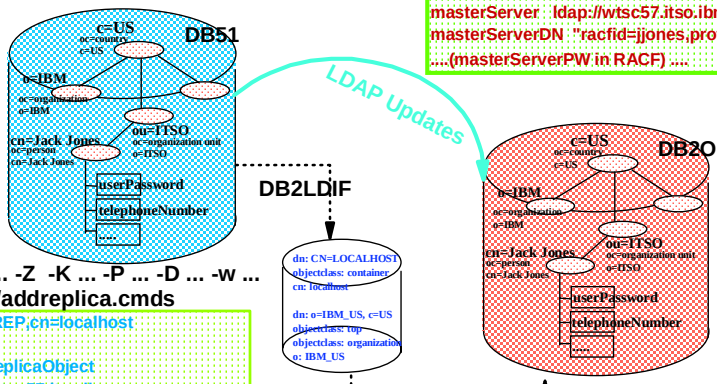
Replication and replicaObject

slapd.conf (master server)

```
include          /etc/ldap/slapd.at.system
include          /etc/ldap/slapd.oc.system
include          /etc/ldap/slapd.at.racf
.....
```

slapd.conf (replica server)

```
include          /etc/ldap/slapd.at.system
include          /etc/ldap/slapd.oc.system
.....
database    rdbm  GLSBRDBM
.....
masterServer   ldap://wtsc57.itso.ibm.com:389
masterServerDN  "racfid=jjones,profiletype=user,sysplex=local"
....(masterServerPW in RACF) ....
```



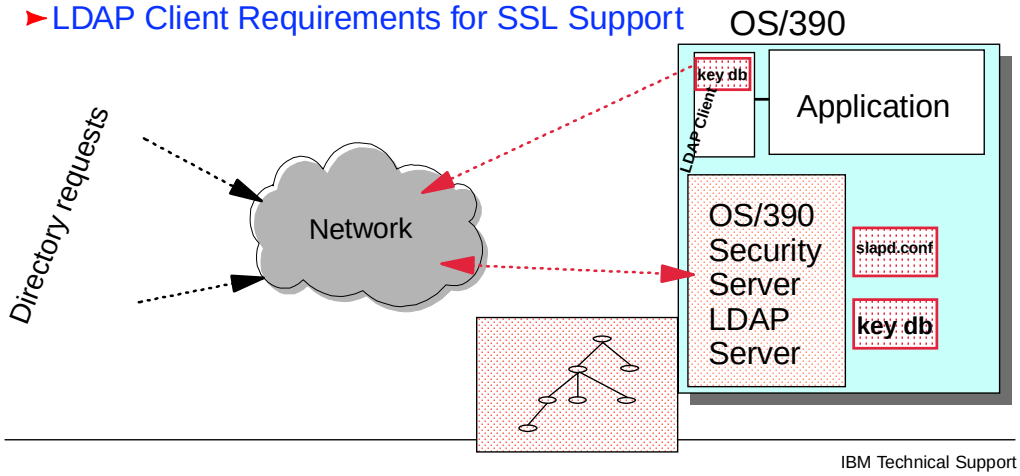
ldapadd -h ... -Z -K ... -P ... -D ... -w ... -f /u/jjones/addreplica.cmds

```
dn:  cn=LDAPREP,cn=localhost
cn:  LDAPREP
objectclass:  replicaObject
replicaHost:  wtsc57.itso.ibm.com
replicaCredentials:  jjpasswd
replicaPort:  1636
replicaUseSSL:  TRUE
description:  "LDAPREP is the backup for LDAPSRV."
replicaBindDN:  racfid=JJONES,profiletype=USER,sysplex=LOCAL
```

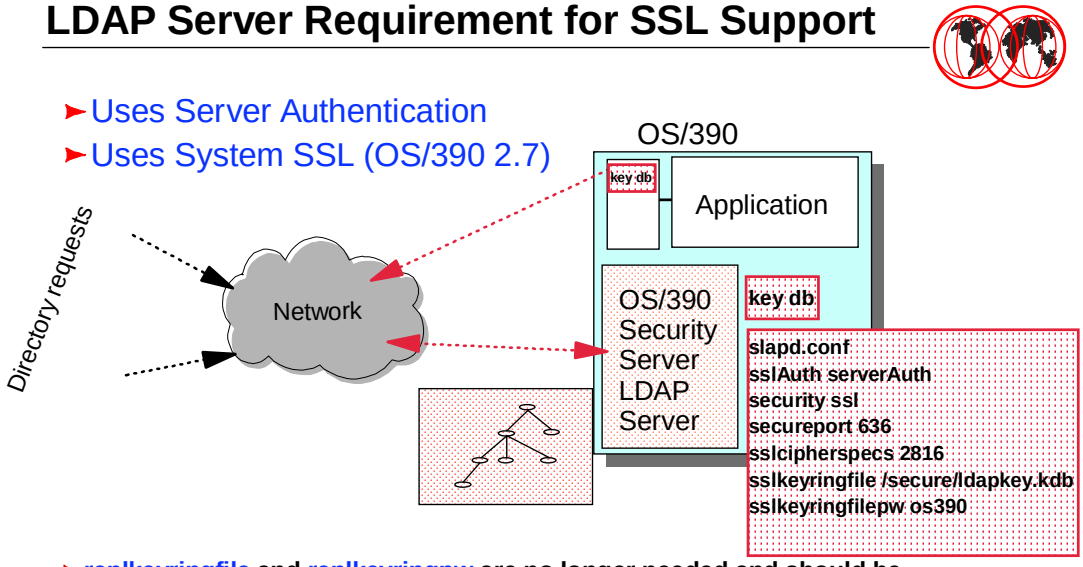
LDIF2DB

Securing the OS/390 LDAP Server with SSL

- LDAP Server Requirement for SSL Support
  - Server Customization
    - Configuration Files
  - LDAP Server Requirements for Key Management
    - Migration Support for MKKF
- LDAP Client Requirements for SSL Support



LDAP Server Requirement for SSL Support



- **replkeyringfile and replkeyringpw are no longer needed and should be removed from slapd.conf**
- **mkkf is not used anymore - to migrate the keyring to new support (System SSL) use: gskkyman -m /secure/oldkeyring.kdb**
- ***OS/390 Cryptographic Services System Secure Sockets Layer Programming Guide - SC24-5877***

IBM Technical Support

- security directive can be:

2. ssl - run in either ssl and non-ssl mode
3. sslonly - run in only ssl mode
4. nossl - run in only non-ssl mode
5. none - run in only non-ssl mode

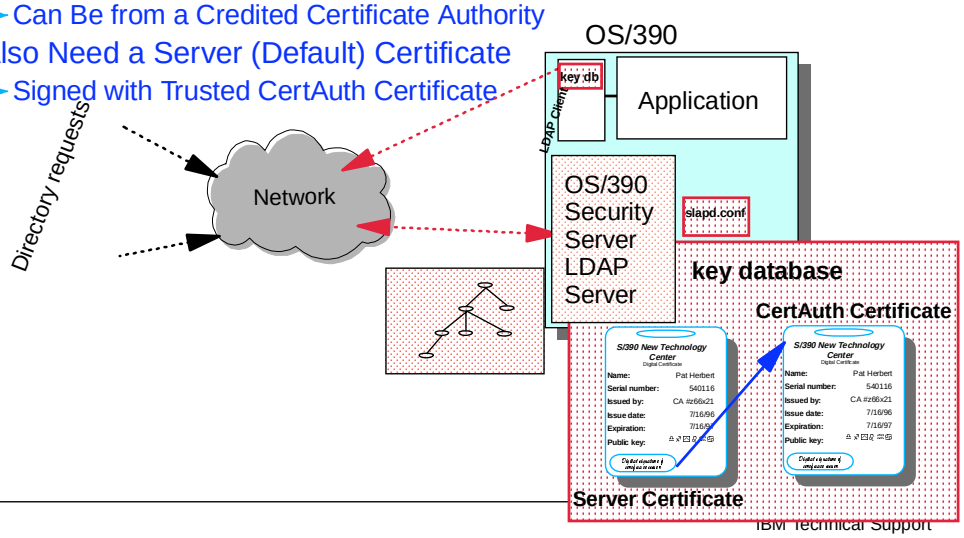
sslcipherspecs directive can be obtained by adding the appropriate values that the LDAP server is suppose to support.

RC4_MD5 (US Strength) 2048
Triple_DES_SHA 256
DES_SHA 512
RC2_MD5 (non-US or Export Strength) .. 4096
RC4_MD5 (non-US or Export Strength) .. 8192

So to support only DES and Triple DES the value of sslcipherspecs would be 768.

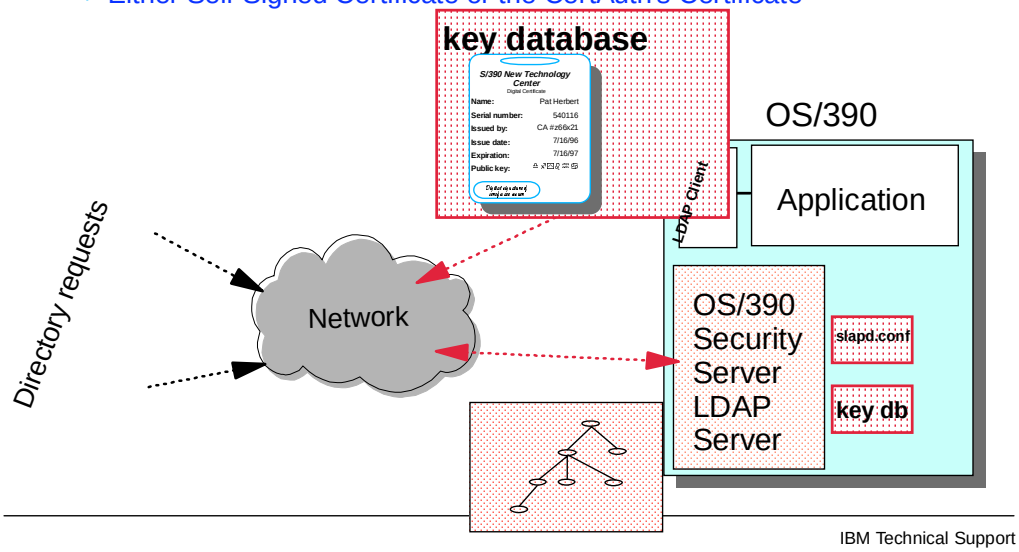
LDAP Server Requirements for Key Mgmt

- Build a Key Database and Fill with Certificates
  - Use gskkyman for Key Management
  - Need a Trusted CertAuth Certificate
    - Can Be Self-Signed Certificate if Appropriate
    - Can Be from a Credited Certificate Authority
  - Also Need a Server (Default) Certificate
    - Signed with Trusted CertAuth Certificate



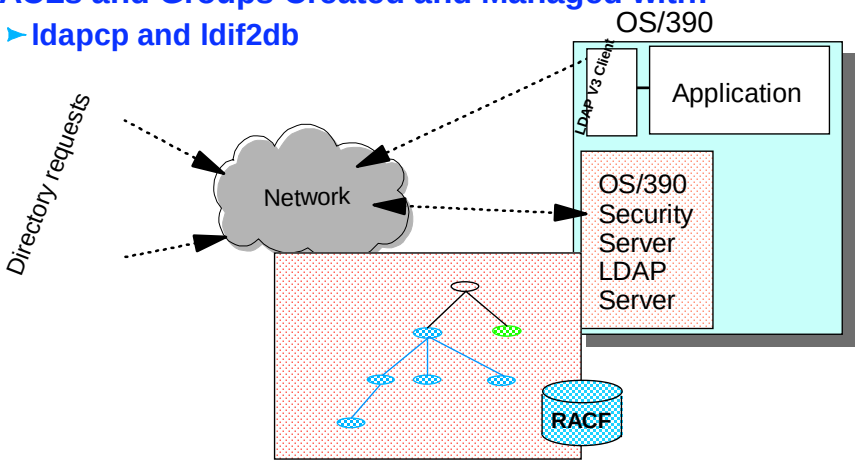
LDAP Client Requirements for SSL Support

- Need a Key Database
  - Used to Verify the LDAP Server's Certificate
  - Must Contain the Signer's Certificate (IMPORT Option)
    - Either Self-Signed Certificate or the CertAuth's Certificate

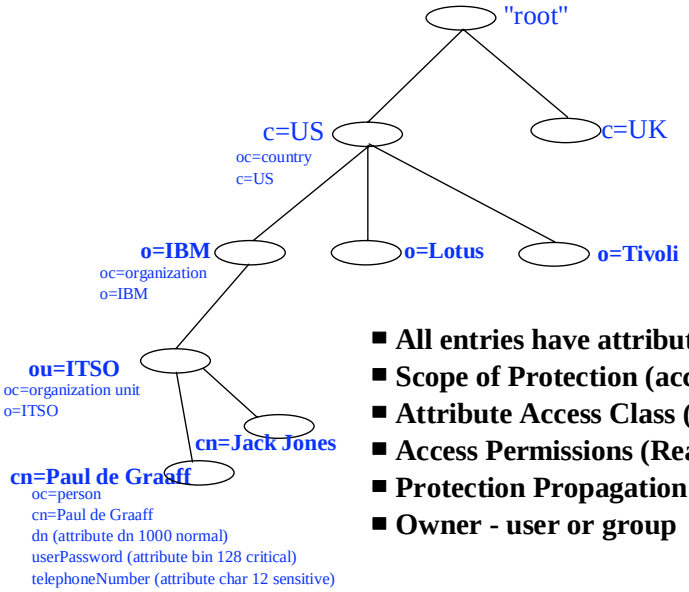


Protecting the Information in the LDAP Server

- ACLs = Access Control Lists
- Control Access to Portions of the Directory or Specific Directory Entries
- Each Directory Entry has DN, Set of Attributes with Values
- ACLs and Groups Created and Managed with:
  - ldapcp and ldif2db



LDAP Directory Content



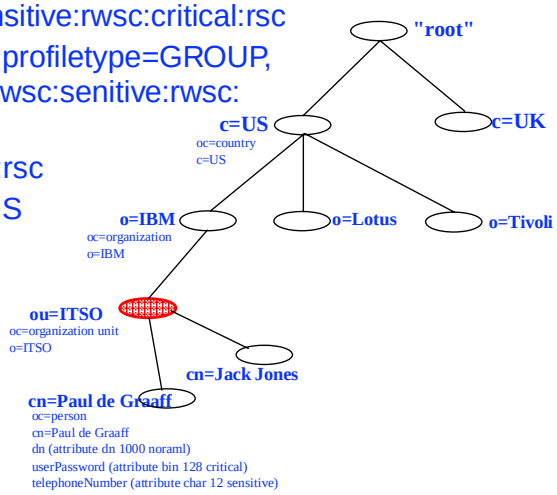
- **All entries have attributes (and values)**
- **Scope of Protection (access-id or group)**
- **Attribute Access Class (normal, sensitive, critical)**
- **Access Permissions (Read, Write, Search, Compare)**
- **Protection Propagation (propagating or overriding)**
- **Owner - user or group**

IBM Technical Support

- Heres a picture of an X.500 Directory model.
-
- Directory is a hierarchy of entries.
- Entries contain attributes.
- Attributes have one or more values.
- An entry's attributes (not their values) are defined by the entry's object class.
- Each entry has a name relative to its parent. This is a relative distinguished name (RDN).
- All RDNs from root to entry put together form a distinguished name (DN).
- RDN: cn=Tim Hahn
- DN: cn=Tim Hahn, o=IBM, c=US

ACL Example 1

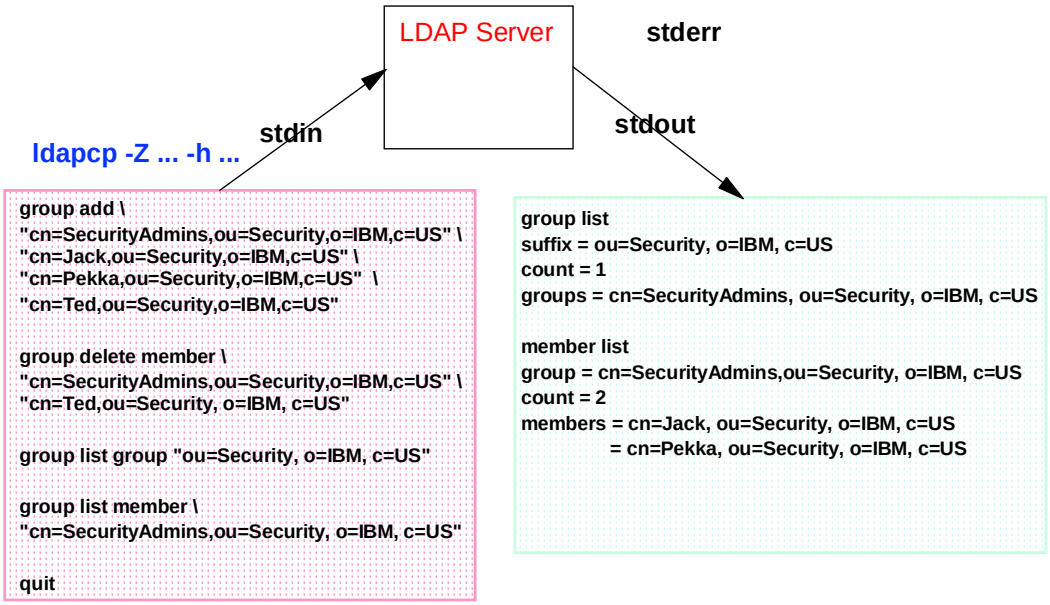
- Protection for: **ou=ITSO, o=IBM, c=US**
  - **aclPropagate:** True
  - **aclEntry:** group=ITSOfolks, o=IBM, c=US:normal:rcs:sensitive:rsc
  - **aclEntry:** access-id:cn=Paul de Graaff, ou=ITSO, o=IBM, c=US:object:ad:normal:rwsc:sensitive:rwsc:critical:rsc
  - **aclEntry:** access-id:racfid=SYS1, profiletype=GROUP, sysplex=local:object:ad:normal:rwsc:senitive:rwsc: critical:wrsc
  - **aclEntry:** group=Anybody:normal:rsc
  - **aclSource:** ou=ITSO, o=IBM, c=US



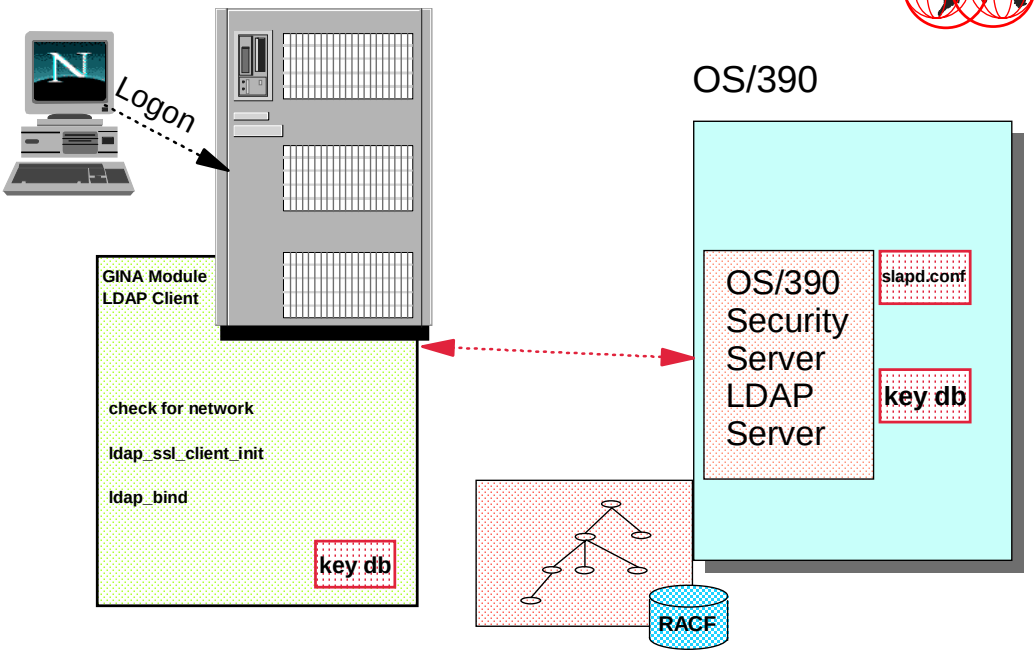
IBM Technical Support

- The Bind-DN can be established with the SDBM, that is, the user can be verified with the RACF database (a RACF userid and password) and the Bind-DN from that RACF userid will be used when checking the ACLs.

Managing Groups with ldapcp



So .... In Theory



For More Information

- **LDAP Protocol**
  - http://www3.innosoft.com/ldapworld/index.html
- **IBM Publications**
  - LDAP Server
    - SC24-5861 OS/390 Security Server LDAP Server Administration and Usage Guide
  - LDAP Client
    - SC28-5878 OS/390 LDAP Client Application Development Guide and Reference
- **HTML shipped in /usr/lpp/ldap/doc**
  - Documentation for both C/C++ LDAP client and JNDI
- **Redbooks**
  - SG24-4986 Understanding LDAP
  - SG24-5158 OS/390 Security Server, Ready for e-business
  - SG24-5110 LDAP Implementation Cookbook
  - SG24-5629 OS/390 Security Server 1999 Updates, Implementation Guide
- **Websites**
  - http://www.s390.ibm.com/security