

ibm.com



e-business

# e-business security for z/OS 1.2

## SecureWay Security Server for z/OS 1.2 Enhancements



**Redbooks**

International Technical Support Organization

© Copyright IBM Corp. 2001

IBM

# Agenda

- **Network Authentication Services (OS/390 Kerberos)**
- **Directory Services on z/OS 1.1**
- **Directory Services Enhancements at z/OS 1.2**
- **RACF Group Size Expansion**
- **SAFTRACE Facility**
- **RACF Support of Mixed Cases Profile Names**

# Kerberos Overview ....

## ☛ Kerberos service (KDC) has 2 parts

- Authentication Service (AS)
  - authenticates users
  - grants TGTs
- Ticket Granting Service
  - Generates session keys
  - Grants service tickets based on TGT

## ☛ Together these are the Key Distribution Center

# Network Authentication and Privacy Services

## Enhancements at z/OS 1.2



**Redbooks**

International Technical Support Organization

# Network Authentication Services Enhancements

- Strong Cryptographic Support
- New Security Server commands and APis
- New and Changed GSS-API Support
- New Security Server Features
- SKRBKDC Automatic Start or Restart

# Network Authentication Services Enhancements - Strong Cryptography

- optional 168 bit key encryption (Triple-DES)
  - ▶ 56 bit or 128 bit preference in krb5.conf
  - ▶ T-DES always available for ticket data, may not be available for user data
- Hardware crypto support
  - ▶ call to ICSF for data encryption (MAXLEN parameter in CSFPRMxx must be large enough for maximum application message size)
- Server changes to handle new key type
  - ▶ Kerberos and GSSAPI APIs
  - ▶ keytab command
  - ▶ KDC krb5.conf and envvar files

# Network Authentication Services Enhancements - Strong Cryptography

- Environment variable SKDC\_TKT\_ENCTYPES  
list of encryption types to be used  
most preferred to least preferred
  - ▶ des-cbc-crc
  - ▶ des-cbc-md4
  - ▶ des-cbc-md5
  - ▶ des-hmac-sha1
  - ▶ des3-cbc-sha1

# Network Authentication Services

## Strong Cryptography - RACF Support

- RACF support for more encryption types for keys

DES, Triple DES, DES with Derivation

- Allow/disallow each type on a per profile basis

Allowed on RDEFINE/RALTER and ADDUSER/ALTUSER

ENCRYPT(DES|NODES  
DES3|NODES3|DESD|NODESD)



# Network Authentication Services

## Strong Cryptography - RACF Support

- RACF support for more encryption types for keys (cont'd)
  - ▶ New support activated by SETROPTS command KERBLVL setting  
**KERBLVL(0|1)**  
Added to SETROPTS command
    - 0 - Process at original level of support (R10/PTF)
    - 1 - Incorporate multiple key functions
- Migration
  - ▶ Do not upgrade to level 1 until all systems sharing the DB have multiple key code level
  - ▶ Can set ENCRYPT values at either level, but has no effect until KERBLVL set to 1

# Network Authentication Services

## Strong Cryptography - RACF Support

### ■ R\_kerbinfo

- ▶ Returns the bits associated with allowable encryption types for this profile
- ▶ Key values return in new three key format

### ■ SMF Unload

- ▶ Unload new SETROPTS KERBLVL value

### ■ Database Templates

- ▶ New ENCRYPT field defined on user and general resource profiles

### ■ Dynamic Parse

- ▶ ENCRYPT keyword added
  - Valid settings: DES|NODES DES3|NODES3  
DESD|NODESD

# New Security Server Commands

- **kpasswd** - change principal's password

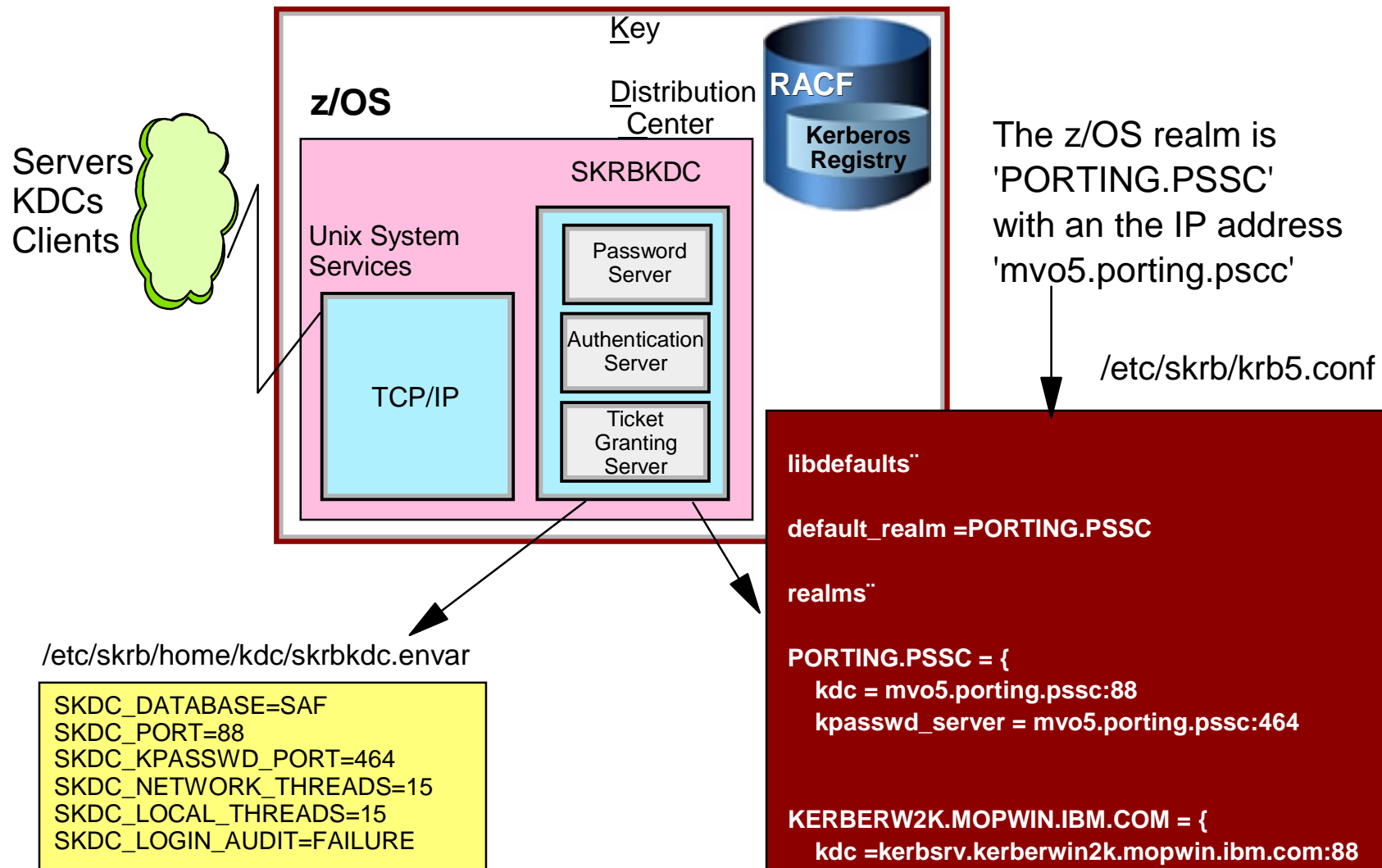
kpasswd [principal]

- ▶ this changes RACF password as well !
- ▶ user must be permitted to the SKRBKDC APPL
- ▶ the password change service is listening to port 464 by default

- **kvno** - Query key version number

kvno [principal]

# z/OS Kerberos Server Setup



# New Security Server Commands

- **kadmin** - this commands allows to manage the Kerberos database on other platforms than z/OS

```
kadmin [-r realm]
        [-p principal]
        [-k keytab]
        [-w password]
        [-A]
        [-e]
```

## Subcommands

```
help [subcommand]
list_principals [expression]
get_principal name
add_principal [options][attributes] name
delete_principal name
modify_principal [options][attributes] name
change_password [-randkey|-pw password] name
rename_principal oldname newname
list_policies [expression]
get_policy name
add_policy [options] name
modify_policy [options] name
delete_policy name
add_key [[-keytab | -k] keytab_name] principal_name
```

- New Kerberos API : kadm5\_

# New Security Server Commands

- New parameters for MODIFY SKRBKDC
  - ▶ Display contents of credentials cache data space  
DISPLAY creds,owner,date
  - ▶ Display active security servers within the sysplex  
DISPLAY XCF
  - ▶ Display list of available encryption types, whether hardware crypto is available, and whether encryption may be used for user data  
DISPLAY CRYPTO
  - ▶ Service level of Kerberos security server  
DISPLAY LEVEL

# New and Changed GSS-API Support

- Import/export credentials and security contexts
- Share credentials among all systems in Sysplex
  - ▶ delegated credentials created on one system may be used to  
init. security context on another system
- Acquire credentials based on specified credentials cache
  - ▶ `gss_krb5_acquire_cred_ccache`

# New Security Server Features

- Dynamic TCP/IP support
  - ▶ Server no longer fails if TCP/IP becomes unavailable
  - ▶ Dynamic addition of TCP/IP network interfaces
    - New TCP/IP stack started
    - New TCP/IP device interface activated
  - ▶ Dynamic removal of TCP/IP network interfaces if unavailable
- Improved DNS lookup
  - ▶ Exploit OS/390-unique statements in /etc/resolv.conf
  - ▶ TCP vs. UDP connections, UDP resolver timeout
  - ▶ Alias keywords to allow use of prefix.TCPIP.DATA
    - (set RESOLVER\_CONFIG)



# SKRBKDC Automatic Start or Restart

- message processing exit to start SKRBKDC
  - ▶ sample in EUVF.SEUVFSAM(STARTKDC)
  - ▶ Sample entry in MPFLSTxx member in SYS1.PARMLIB:  
`BPXI004I, SUP(NO), USEREXIT(STARTKDC)`
- ARM automatic restart
  - ▶ element type is SYSKERB - restart level 2.
  - ▶ element name is EUVFKDC\_sysname
    - name for the SKRBKDC started task on system DCESEC4 would be EUVFKDC\_DCESEC4

# Miscellaneous

- Sample applications
  - ▶ Kerberos message functions
    - [/usr/lpp/skrb/examples/krbmsg\\_test](/usr/lpp/skrb/examples/krbmsg_test)
    - client, server, README
  - ▶ GSS-API
    - [/usr/lpp/skrb/examples/gssapi\\_test](/usr/lpp/skrb/examples/gssapi_test)
    - client, server, delegate, README
- Migration utility : kmigrate
  - ▶ utility to help prime Kerberos registry
    - from DCE
    - from RACF
  - ▶ Generates ADDUSER and ALTUSER commands
  - ▶ Free Download from  
<http://www.s390.ibm.com/racf/goodies.html>

# Directory Services Enhancements in z/OS 1.2



**Redbooks**

International Technical Support Organization

# z/OS V1R2 LDAP Content

- Miscellaneous changes
  - ▶ LDAP Server executables must now be always APF authorized
    - PDS <GLDHLQ>.SGLDLNK
    - GLDCLDAP in HFS
- New .ldif files
  - /usr/lpp/ldap/etc/EntrustPKIV4.ldif, EntrustPKIV5.ldif
  - /usr/lpp/ldap/etc/RFC2587.ldif
  - /usr/lpp/ldap/etc/MS.ActiveDirectory.ldif
  - /usr/lpp/ldap/etc/SecurityIdentities.ldif
  - /usr/lpp/ldap/etc/RACF.2.ldif
  - /usr/lpp/ldap/etc/NativeAuthentication.ldif
- Files added via these PTFs to OS/390 R10
  - LDAP Configuration Utility (OW47594)
  - Native Authentication (OW47596)

# Complete List of Schema Modules

- ChangeLog.Idif
- CommServer.Idif
- ComponentBroker.Idif
- DB2.Idif
- DMTF.Idif
- EntrustPKIV4.Idif
- EntrustPKIV5.Idif
- IBM.Idif
- Kerberos-V1.Idif
- ManagedSystemInfrastructure.Idif
- MCI.Idif
- MetaDirectory.Idif
- MS.ActiveDirectory.Idif
- NativeAuthentication.Idif
- Netscape.Idif
- Netscape-V2.Idif
- NFI.Idif
- nisSchema.Idif
- OnDemandServer.Idif
- OtherStandard.Idif
- PolicyDirector.Idif
- RACF.Idif
- RACF.2.Idif
- Registered Software.Idif
- RFC2252.Idif
- RFC2256.Idif
- RFC2587.Idif
- RFC2713.Idif
- RFC2714.Idif
- SecurityIdentities.Idif
- System.Idif
- System-V2.Idif
- UniversalMessaging.Idif
- UNIX.Idif
- WebSphereNaming.Idif
- X.520.Idif

# **z/OS V1R2 LDAP Content**

- **LDAP C/C++ Client enhancements**
  - ▶ **DNS Server locate**
  - ▶ **Search result caching**
  - ▶ **Kerberos authentication**
  - ▶ **Extended operations**
  
- **LDAP Server enhancements**
  - ▶ LDAP configuration utility
  - ▶ Server front-end performance/scalability
  - ▶ SDBM Enhancements
  - ▶ Native Authentication
  - ▶ Kerberos Authentication

# z/OS 1.2 LDAP Client Enhancements

## Locating LDAP server information

- New client C/C++ APIs to perform the following operations:
  - Obtain LDAP server information published in the Domain Name System (or DNS).
  - Obtain LDAP server information from a local configuration file.
  - Store LDAP server information obtained from DNS into a local configuration file.
- Client applications can omit the explicit hostname/port information in the LDAP URL; this information can be obtained through the new API
- Commonality with SecureWay directory clients on other platforms

# z/OS 1.2 LDAP Client Enhancements

## Caching LDAP search results

- New client environment variables to create a single client-side cache (Global Cache), and new C/C++ APIs to create a client-side cache associated with a specific LDAP connection
- Subsequent `ldap_search` requests using the same search constraints (host, port, search base, authenticated bind dn, filter, etc...) can be satisfied from the cache
- Global Caching does NOT require modification and recompilation of existing OS/390 LDAP applications



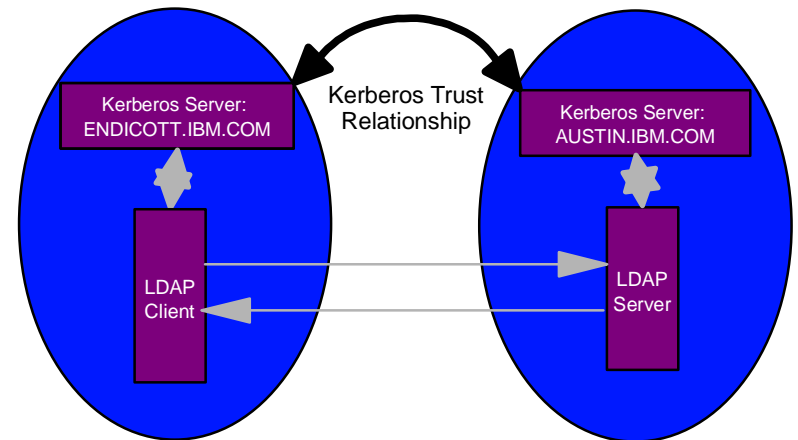
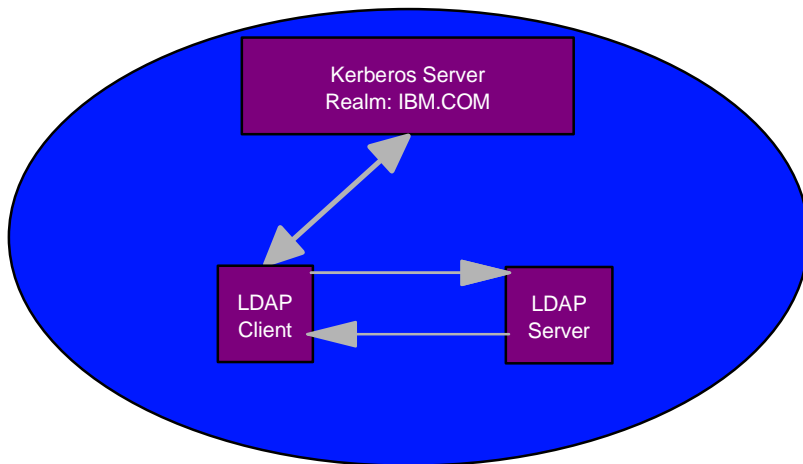
# **z/OS 1.2 LDAP Client Enhancements**

## **LDAP C/C++ Client - Kerberos Support**

- Bind to any LDAP Server that supports Kerberos Version 5 binds (z/OS, AIX, Active Directory)
  - ▶ Secure Authentication
  - ▶ Single-Signon Environment
  - ▶ Mutual Authentication between Client and Server
- Kerberos DLL (EUVFKDLL) must be in STEPLIB, LIBPATH, LPALIB or LINKLIST
- ldap\_sasl\_bind\_s() C/C++ API
  - ▶ LDAP\_MECHANISM\_GSSAPI
  - ▶ Must obtain a valid Kerberos Ticket Granting Ticket (TGT) prior to invoking the API (kinit command or gss\_acquire\_cred())

# LDAP C/C++ Client - Kerberos Support

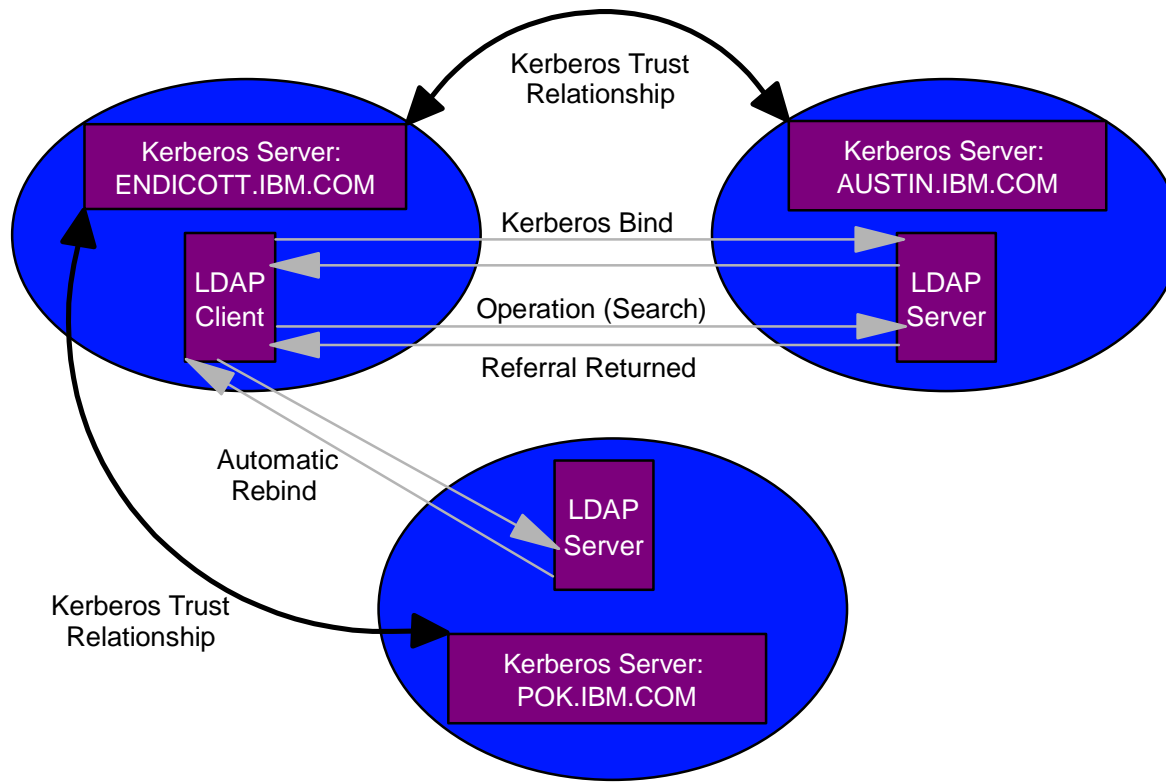
- All of the LDAP utility programs (ldapsearch, etc.) have been updated to perform Kerberos Version 5 binds as well as Kerberos Authenticated Referrals
- The LDAP utility programs will only use the default credentials for the current login context.
  - ▶ ldapsearch -h host -p port **-V 3 -S GSSAPI** -b base filter



# z/OS 1.2 LDAP Client Enhancements

## LDAP C/C++ Client - Kerberos Support

- Kerberos Authenticated Referrals are supported
  - ▶ if last bind method was Kerberos
- Kerberos option for delegation is supported



# z/OS 1.2 LDAP Client Enhancements

## Extended Operation Client API's

- Extended operations are defined for special purposes and may perform several LDAP operations before returning a result
  - ▶ Operation to perform identified by OID
  - ▶ Input/output data and format is defined by the documentation for the OID
- At this time, z/OS LDAP Server does not support any extended operations. Will return LDAP\_PROTOCOL\_ERROR as defined by LDAP RFC
- Support provided for extended operation APIs in the z/OS LDAP client for LDAP applications to access extended operations supported by other LDAP Servers, IBM and non-IBM
  - ▶ ldap\_extended\_operation
  - ▶ ldap\_extended\_operation\_s
  - ▶ ldap\_parse\_extended\_result

# V1R2 LDAP Content

- LDAP C/C++ Client enhancements
  - ▶ DNS Server locate
  - ▶ Search result caching
  - ▶ Kerberos authentication
  - ▶ Extended operations
  
- **LDAP Server enhancements**
  - ▶ **LDAP configuration utility**
  - ▶ **Server front-end performance/scalability**
  - ▶ **SDBM Enhancements**
  - ▶ **Native Authentication**
  - ▶ **Kerberos Authentication**

# LDAP Configuration Utility



**Redbooks**

International Technical Support Organization

# z/OS 1.2 LDAP Server Enhancements

## LDAP Configuration Utility Overview

### ■ **ldapcnf**

- ▶ z/OS Unix utility to assist customer for z/OS LDAP Server configuration
- ▶ Generates
  - JCL jobs to accomplish the updates of all the z/OS components
  - Configuration files necessary for server to operate
- ▶ Customer input into only one file for simple configuration, three additional files allow for complex configuration input
  - Removes need for redundant updates
- ▶ Establishes and segregates component updates
  - Generates jobs executed by user with proper authority for the different components
- ▶ Available for OS/390 2.10 via PTF (OW47594)

# z/OS 1.2 LDAP Server Enhancements

## ldapcnf Invocation and outputs

- **ldapcnf** is invoked by:

- /usr/lpp/ldap/sbin/ldapcnf -i *ldap.profile*

- **ldap.profile**

- environment variable file that the customer must update before invoking ldapcnf. File shipped in /usr/lpp/ldap/config

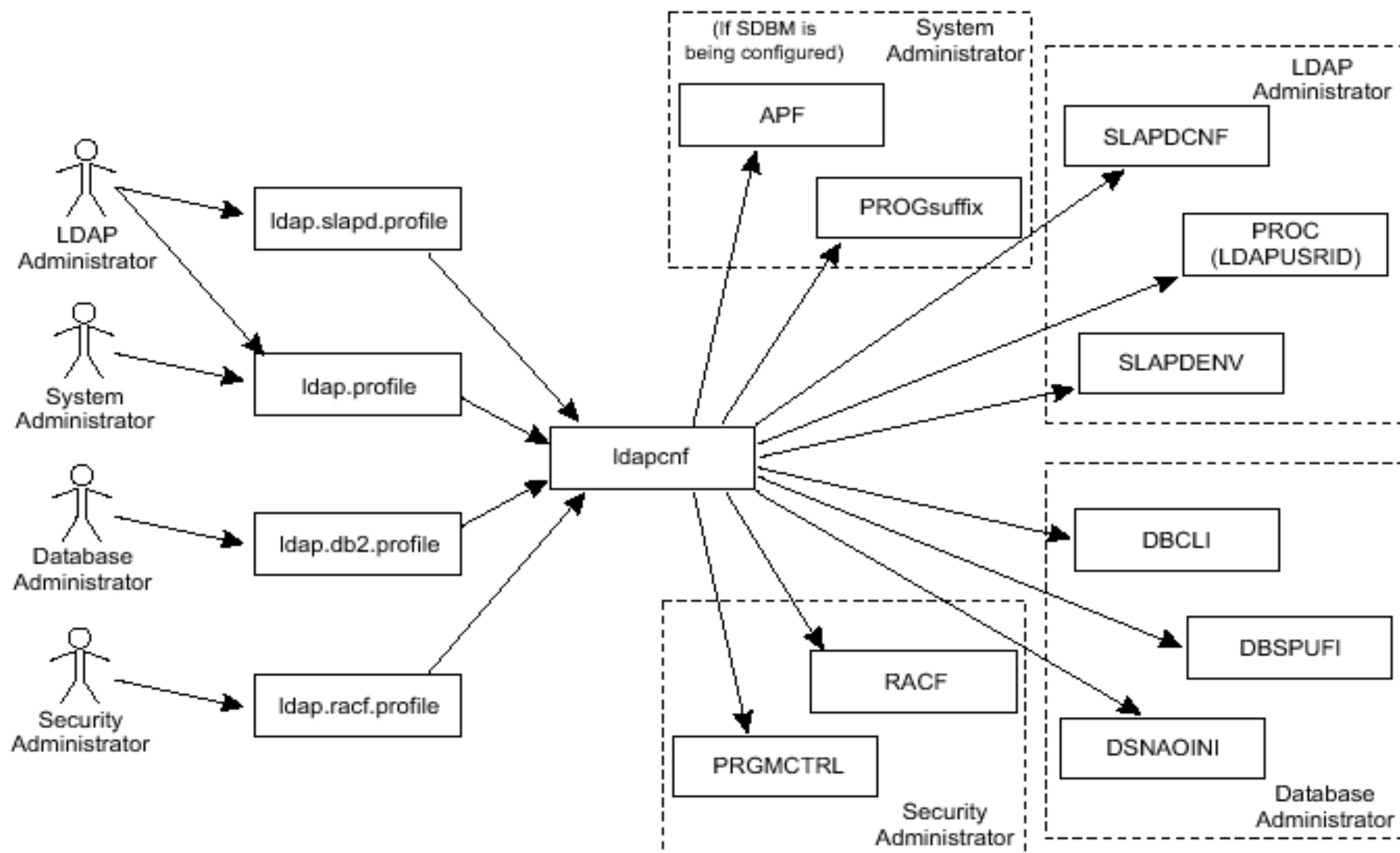
- **Outputs**

- ▶ Error messages if required variables are not assigned values in the ldap.profile file or fail simplified syntax checking
  - ▶ Warning messages if overwriting existing output data set
  - ▶ Multiple JCL jobs that will update the various z/OS components
  - ▶ PROG member to establish APF Authorization
  - ▶ Configuration files for various components
  - ▶ PROC to start the server



# z/OS 1.2 LDAP Server Enhancements

## Idapcnf Overview



# LDAP Server Front End Performance/Scalability



**Redbooks**

International Technical Support Organization

# LDAP Frontend Performance Interactions and Dependencies

- Support of increased number of concurrent clients' connections : approximately up to 65500
- Can now listen on a specific IP address
- Can now listen on multiple secure and non-secure ports for incoming requests (as opposed to one non-secure and one secure port only)
- New configuration parameters
  - ▶ commThreads
  - ▶ listen
  - ▶ idleConnectionTimeout

# LDAP Frontend Performance Interactions and Dependencies

- New LDAP configuration parameters in slapd.conf
  - ***commThreads***  
Number of threads initialized when the LDAP server is first invoked.  
Default: 10
  - ***listen***  
allows for the LDAP server to bind and listen for client requests on a particular IP address or hostname and a port number. Default : listen on any available IP address on port 389 (non-secure port)
  - ***idleConnectionTimeout***  
Length of time in seconds that the LDAP server will wait on an idle client connection. Default: 0 (indefinite)
- Migration and Coexistence
  - ***maxThreads*** and ***waitingThreads*** parameters in the configuration file are now ignored (message issued)
  - ***security***, ***port***, and ***securePort*** parameters being deprecated

# LDAP Frontend Performance Interactions and Dependencies

- Examples of new ***listen*** configuration option
  - ▶ `listen ldaps://:500`
  - ▶ `listen ldap://:400`
  - ▶ `listen ldap://us.endicott.ibm.com:777`
  - ▶ `listen ldaps://9.130.77.27:999`
- specify the option multiple times for multiple listen directives
  - ▶ `slapd -l ldap://:389 -l ldaps://:636`
- LDAP Server command line option changes due to use of listen:
  - ▶ `-h host`, `-p port` and `-s secureport` are deprecated
  - ▶ `-l listen` replaces the options above and supports a format similar to the listen configuration parameter (ie., an LDAP url)

# LDAP Server SDBM Enhancements



**Redbooks**

International Technical Support Organization

# LDAP Server

## Overview of SDBM Enhancements

- additional user segments managed through LDAP
  - ▶ LNOTES segment - short name
  - ▶ NDS segment - user name
  - ▶ KERB segment - kerberos name, maximum ticket life, key version
- new application id search filters to search for a RACF user or group with a specific identity  
LNOTES, NDS, KERB name, OMVS UID and OMVS GID
- users connections to RACF groups managed through LDAP
  - ▶ add and remove a user from a group
  - ▶ display and modify connection information

# LDAP server

## Additional User Segments

- new object classes and attributes for 3 RACF user segments:

- ▶ LNOTES    objectclass racfLNotesSegment  
             requires objectClass  
             allows  
             racfLNotesShortName

- ▶ NDS       objectclass racfNDSegment  
             requires objectClass  
             allows racfNDSUserName

- ▶ KERB      objectclass racfKerberosInfo  
             requires objectClass  
             allows racfCurKeyVersion,  
                    krbPrincipalName,  
                    maxTicketAge

racfLNotesShortName, racfNDSUserName and KrbPrincipalName (except for the REALM part) are case-sensitive



# LDAP Server

## Additional User Segments Example

### ■ SDBM user entry

```
ldapsearch . . . -b racfid=t2,profiletype=user,sysplex=myplex "objectclass=*"
```

```
racfid=T2,profiletype=USER,sysplex=myplex
```

```
. . .
```

```
objectclass=racfKerberosInfo
```

```
objectclass=racfLNotesSegment
```

```
objectclass=racfNDSSegment
```

```
. . .
```

```
krbprincipalname=myKerbName@DCESET3.ENDICOTT.IBM.COM
```

```
maxticketage=0000000200
```

```
racfcurkeyversion=001
```

```
racflnotesshortname=myShortName
```

```
racfndsusername=myUserName
```

```
. . .
```

# LDAP Server

## New Application ID Search Filters

- simple filters only:
  - ▶ `racfLNotesShortName=value`
  - ▶ `racfNDSUserName=value`
  - ▶ `krbPrincipalName=value`
  - ▶ `racfOmvsUid=value`
  - ▶ `racfOmvsGroupId=value`
- valid for search starting at
  - ▶ root base, with subtree scope
  - ▶ base is `profiletype=user` for all except `racfOmvsGroupId`
  - ▶ base is `profiletype=group` for `racfOmvsGroupId` only
- for LNOTES, NDS, and KERB filters, server must have read access to IRR.RUSERMAP resource in FACILITY class

# LDAP Server

## New Application ID Search Filters Examples

### ■ Search for user with a given NDS user name

```
ldapsearch . . . -b profiletype=USER,sysplex=myplex "racfNDSUserName=myUserName"
```

```
racfid=T2,profiletype=USER,sysplex=myplex
```

```
objectclass=racfUser
```

```
objectclass=racfNDSegment
```

```
. . .
```

```
racfid=T2
```

```
racfndsusername=myUserName
```

```
. . .
```

### Search for user with a given LNOTES short name

```
ldapsearch . . . -b sysplex=myplex "racflnotesshortname=myshortname"
```

```
/* no results since racflnotesshortname is case-sensitive and value was myShortName */
```

### Search for user with a given OMVS uid

```
ldapsearch . . . -b sysplex=myplex "racfomvsuid=123456"
```

```
racfid=T2,profiletype=USER,sysplex=myplex
```

```
objectclass=racfUser
```

```
objectclass=racfUserOmvsSegment
```

```
. . .
```

```
racfid=T2
```

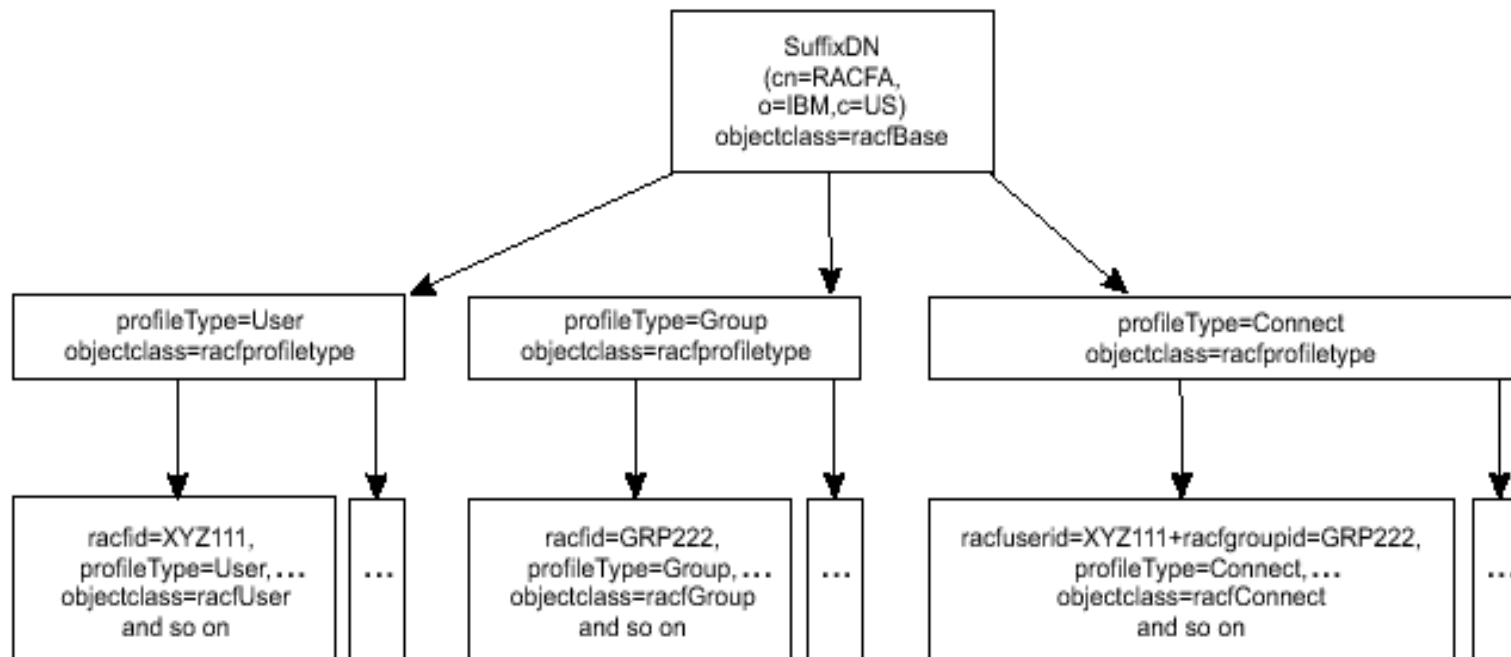
```
racfomvsuid=0000123456
```

```
. . .
```

# LDAP Server

## Managing User-Group Connections

- Can add/remove a user from a group and can display/modify information controlling a user's membership in a group using LDAP commands
- new SDBM subtree for connect entries
  - ▶ profiletype=connect, *suffix*
  - ▶ compound RDN to specify user + group



# LDAP Server

## Managing User-Group Connections (cont)

- Connect entry attributes represent connection information

objectclass racfConnect

requires objectClass,  
    racfuserid,  
    racfgroupid

allows

    racfConnectAttributes,  
    racfConnectAuthDate,  
    racfConnectCount,  
    racfConnectGroupAuthority,  
    racfConnectGroupUACC,  
    racfConnectLastConnect,  
    racfConnectOwner,  
    racfConnectResumeDate,  
    racfConnectRevokeDate

# LDAP Server

## Managing User-Group Connections (cont)

### **SDBM add a connection (i.e. add an existing user to an existing group)**

ldif\_add\_file:

```
dn: racfuserid=T3+racfgroupid=GRP1,profiletype=connect,sysplex=myplex
objectclass=racfConnect
racfuserid=T3
racfgroupid=GRP1
racfconnectgroupauthority=USE
racfconnectowner=racfid=T2,profiletype=USER,sysplex=myplex
racfconnectgroupuacc=ALTER
racfconnectattributes=SPECIAL OPERATIONS
```

> ldapadd . . . -f ldif\_add\_file

adding new entry racfuserid=T3+racfgroupid=GRP1,profiletype=CONNECT,sysplex=myplex

### **SDBM modify a connection**

ldif\_mod\_file:

```
dn: racfuserid=T3+racfgroupid=GRP1,profiletype=connect,sysplex=myplex
changetype: modify
add: x
racfconnectrevokedate: 09/20/02
```

> ldapmodify . . . -f ldif\_mod\_file

modifying entry racfuserid=T3+racfgroupid=GRP1,profiletype=CONNECT,sysplex=myplex

### **SDBM delete a connection (i.e. remove a user from a group)**

> ldapdelete . . . "racfuserid=T3+racfgroupid=GRP1,profiletype=CONNECT,sysplex=myplex "

# LDAP Server

## Managing User-Group Connections (cont)

- connect entry search: AND filter to specify user and group

`(&(racfuserid=user)(racfgroupid=group))`

*user* and *group* can contain RACF wildcards (%,\* )

- returns DN of connect entries with a user matching *user* who is connected to a group matching *group*

- two shorthand versions of filter supported:

`racfuserid=user` same as `(&(racfuserid=user)(racfgroupid=*))`

- ▶ returns a DN for each of *user*'s connections

`racfgroupid=group` same as `(&(racfuserid=*)(racfgroupid=group))`

- ▶ returns a DN for each connection to *group*

# LDAP Server

## Managing User-Group Connections (cont)

**SDBM search for the connection entry DNs for each group of which user T2 is a member**

```
/* same as "(&(racfuserid=t2)(racfgroupid=*))" filter */  
> ldapsearch . . . -b profiletype=CONNECT,sysplex=myplex "racfuserid=t2"  
racfuserid=T2+racfgroupid=SYS1,profiletype=CONNECT,sysplex=myplex  
racfuserid=T2+racfgroupid=DCEMVS,profiletype=CONNECT,sysplex=myplex
```

**SDBM search for the connection entry DNs for each member of group DCEMVS**

```
/* same as "(&(racfuserid=*)(racfgroupid=DCEMVS))" filter */  
> ldapsearch . . . -b profiletype=CONNECT,sysplex=myplex "racfgroupid=DCEMVS"  
racfuserid=G12345+racfgroupid=DCEMVS,profiletype=CONNECT,sysplex=myplex  
racfuserid=U1+racfgroupid=DCEMVS,profiletype=CONNECT,sysplex=myplex  
racfuserid=T2+racfgroupid=DCEMVS,profiletype=CONNECT,sysplex=myplex  
racfuserid=G13579+racfgroupid=DCEMVS,profiletype=CONNECT,sysplex=myplex
```

**SDBM search for the connection entry DNs for each user with a name starting with G who is a member of a group with a name starting with DCE**

```
> ldapsearch ... -b profiletype=CONNECT,sysplex=myplex "(&(racfuserid=G*)(racfgroupid=DCE*))"  
racfuserid=G12345+racfgroupid=DCEMVS,profiletype=CONNECT,sysplex=myplex  
racfuserid=G13579+racfgroupid=DCEMVS,profiletype=CONNECT,sysplex=myplex  
racfuserid=GXFRED+racfgroupid=DCEXGRP,profiletype=CONNECT,sysplex=myplex  
racfuserid=GTEMP+racfgroupid=DCE001,profiletype=CONNECT,sysplex=myplex
```



# LDAP Server TDBM Native Authentication



**Redbooks**

International Technical Support Organization

# LDAP Server

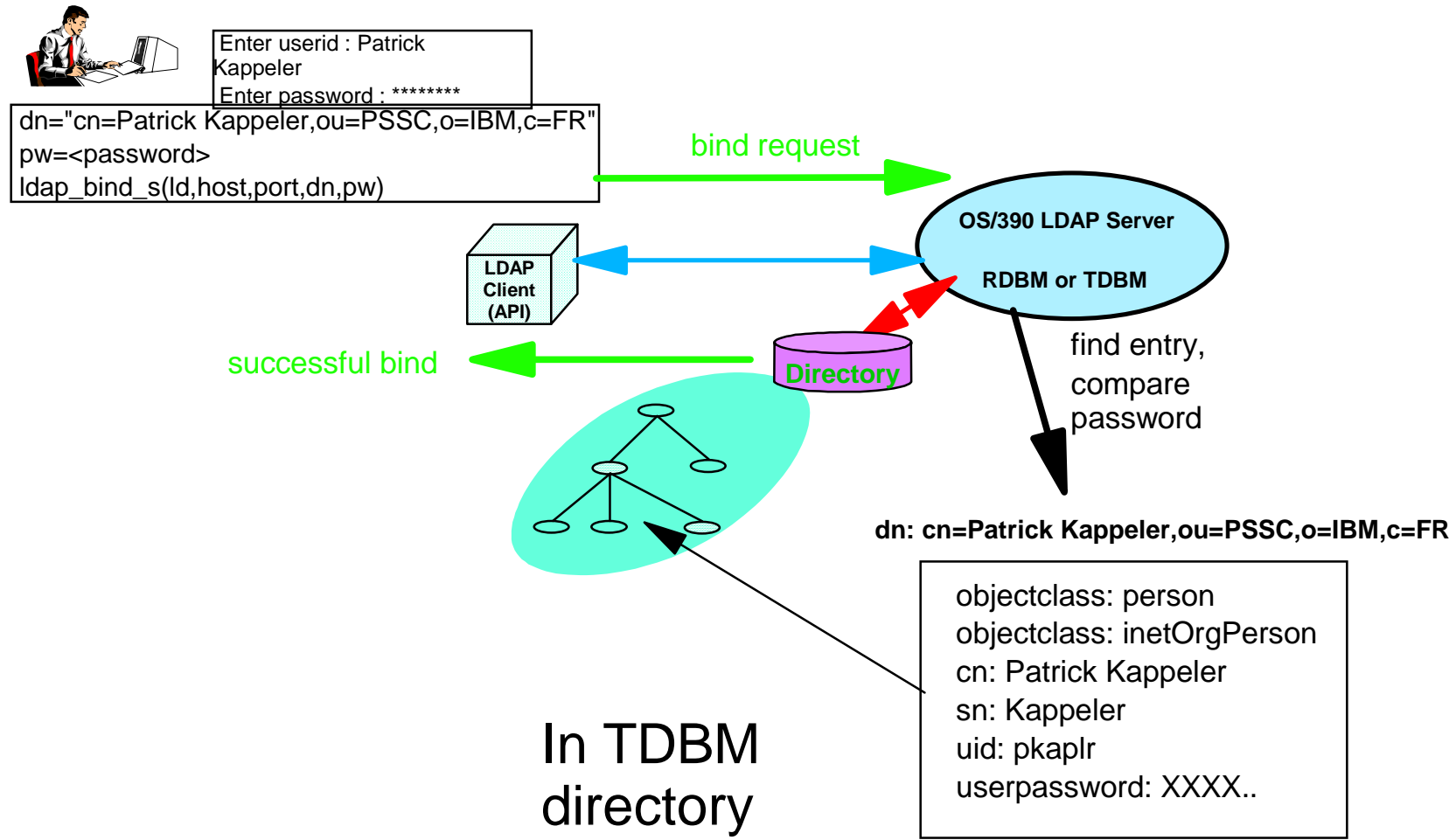
## TDBM Native Authentication

- Provide support in the TDBM backend to bind with a password that is passed to and verified by RACF
  - ▶ Provide a way to map RACF IDs to LDAP entries so that the appropriate information can be passed to the Security Server for password authentication

# LDAP Server

## TDBM Native Authentication

### Authenticated Bind, without native authentication



# LDAP Server

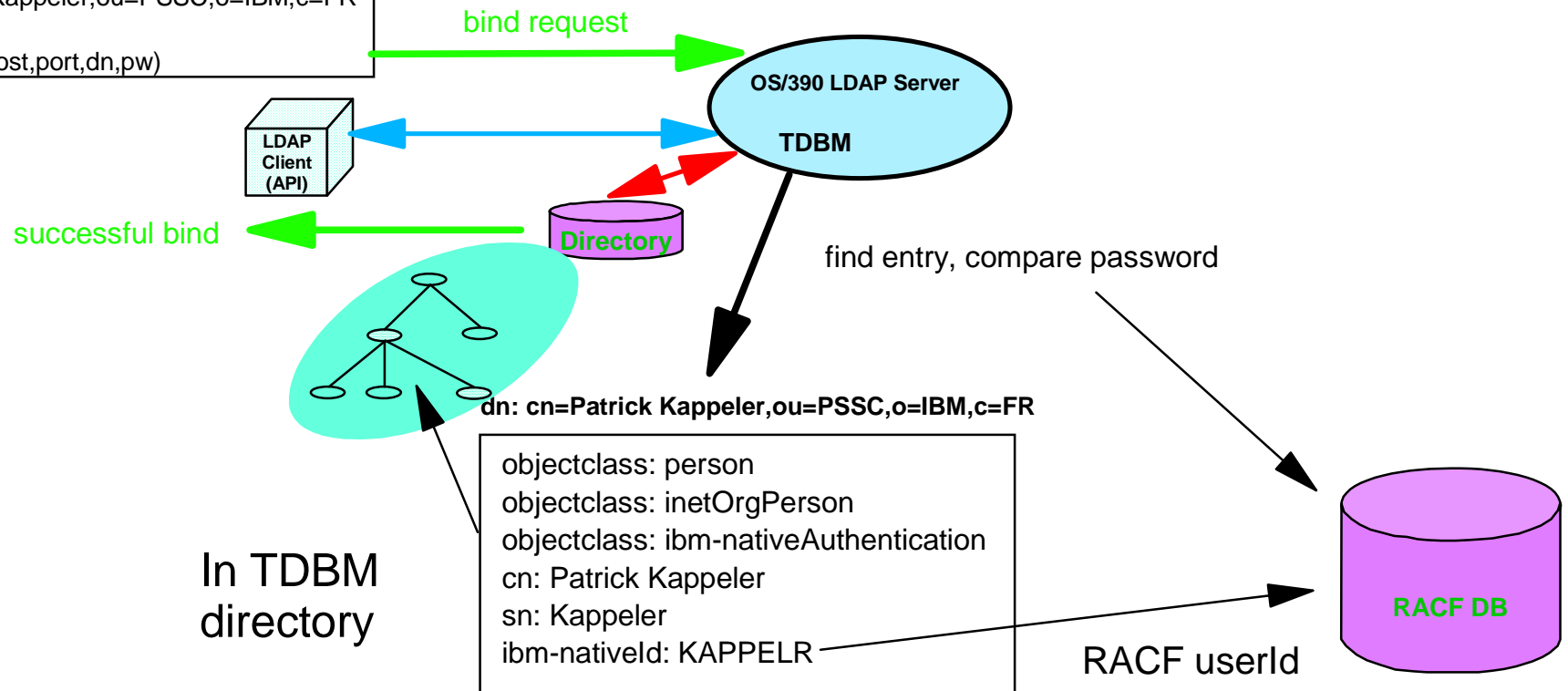
## TDBM Native Authentication

### Authenticated Bind, with native authentication



Enter userid : Patrick Kappeler  
Enter password : \*\*\*\*\*

dn="cn=Patrick Kappeler,ou=PSSC,o=IBM,c=FR"  
pw=<password>  
ldap\_bind\_s(ld,host,port,dn,pw)



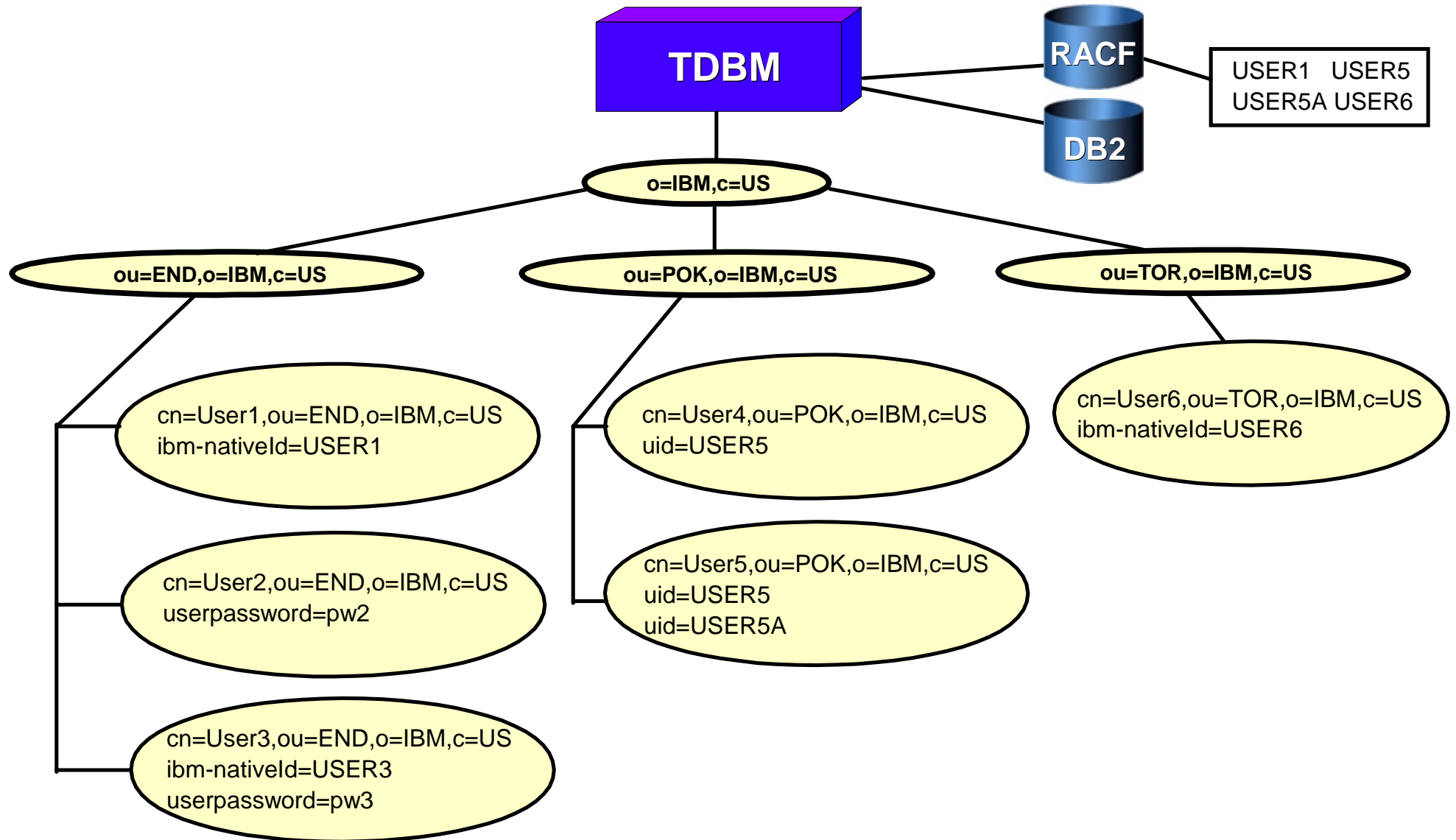
# LDAP Server

## TDBM Native Authentication

- Must load the NativeAuthentication.Idif schema file
  - ▶ New **attribute ibm-nativeId** : specifies the RACF ID associated with this entry
  - ▶ New Objectclass **ibm-nativeAuthentication**
- Entries that are added and subject to Native Authentication cannot contain the userpassword attribute
- RACF group gathering will not be performed as a part of authentication

# LDAP Server

## Native Authentication



# LDAP Server

## Native Authentication Configuration Options

### ■ TDBM Section

#### ► nativeAuthSubtree <all|DN>

**all** - the entire TDBM directory will use Native Authentication

**DN** - subtree that contains entries that will use Native Authentication

#### ► useNativeAuth <selected|all|off>

**selected** - entries located in native subtrees that contain the **ibm-nativeId** attribute

**all** - every entry in native subtrees will use native authentication with **ibm-nativeId** or **uid** attribute

**off** - Native Authentication is disabled

# LDAP Server

## Native Bind

- Notes:
  - ▶ If a multi-valued UID is detected the operation will fail since the server does not know which value to use.
  - ▶ If Native Authentication fails due to the fact that the RACF ID is not defined then a regular LDAP bind will be attempted.



# LDAP Server

## Native Authentication Configuration Options

- Allow native (RACF) passwords to be changed via an LDAP modify to the TDBM directory for entries where native authentication applies
  - ▶ TDBM Section
    - nativeUpdateAllowed <on|yes|off|no>
      - on|yes** - update of the native password is allowed
      - off|no** - not allowed to update your native password
  - ▶ Issue a modify/delete with the old password followed by
    - a modify/add of the new password
    - userpassword=<oldracfpassword>
    - +userpassword=<newracfpassword>

# LDAP Server Kerberos Authentication



**Redbooks**

International Technical Support Organization

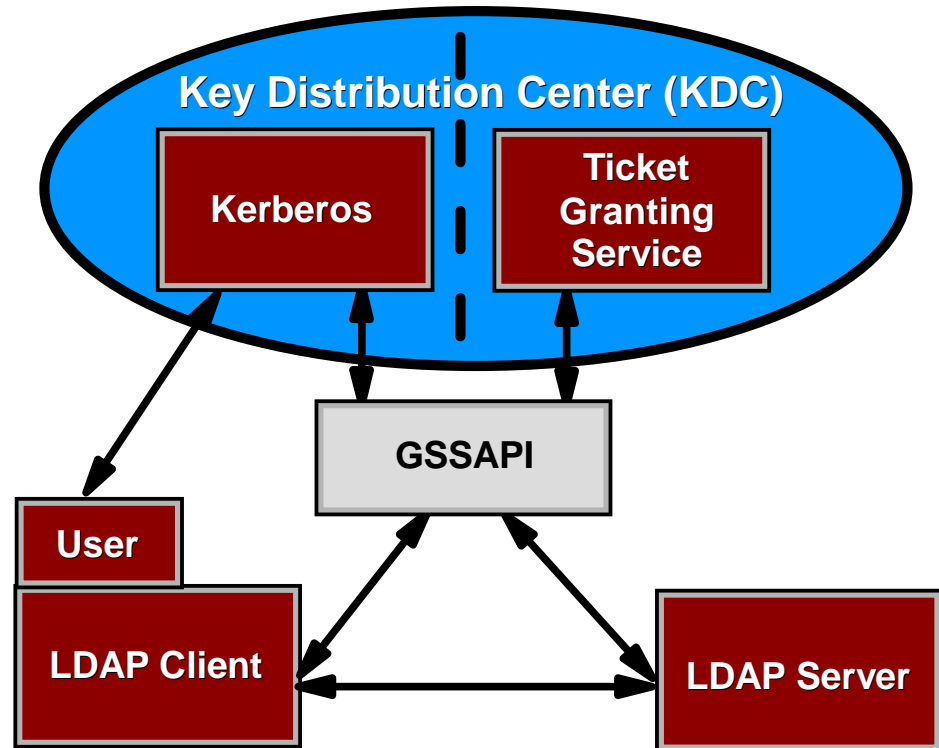
# LDAP Server-Side Kerberos Bind

- Provide support for GSSAPI SASL binds and secure authentication using Kerberos ticket
- Publish Kerberos information in the servers rootDSE  
supportedsaslmecanism=GSSAPI  
ldapservicename=hostname@REALM
- Kerberos is used only for LDAP authentication
  - ▶ No support for Kerberos integrity and confidentiality options
- LDAP access control is performed on the basis of the LDAP distinguished name

# LDAP Kerberos Configuration

- Install and configure the z/OS Network Authentication and Privacy Service Kerberos on the machine where the LDAP server will run.
- Create the LDAP servers Kerberos KDC account.
- Optionally generate the servers keytab file (if not on the same system as the KDC)
- Specify the necessary Kerberos options in the slapd.conf file

GLD0170I Kerberos authentication support has been enabled.  
GLD0171I Kerberos authentication support has NOT been enabled  
GLD0172E Dynamic load of Kerberos DLL failed  
GLD0173E Server was unable to acquire Kerberos credentials



# LDAP Kerberos Configuration Options

# Global Section


**supportKrb5** yes

**serverKrbPrinc** LDAP/myhost@MYREALM.COM

**krbLDAPAdmin** ibm-kn=ldapadm@MYREALM.COM

**krbKeytab** none

use RACF DB as keytab  
(must be on the same z/OS as  
RACF DB)




# TDBM Section

**krbIdentityMap** on

# SDBM Section

**krbIdentityMap** on

map Kerberos principal to  
backend DN for access  
control



# LDAP Kerberos Identity Mapping Schema

- New schema files that must be loaded into the LDAP Server to enable GSSAPI authentication :
  - ▶ MS.ActiveDirectory.Idif and SecurityIdentities.Idif
- **Attributetypes**
  - ▶ krbRealmName-V2
  - ▶ krbPrincSubtree
  - ▶ krbPrincipalName
  - ▶ krbAliasedObjectName
  - ▶ krbHintAliases
  - ▶ altSecurityIdentities
  - ▶ ibm-kn or ibm-kerberosName
- **Objectclasses**
  - ▶ krbRealm-V2
  - ▶ ibm-securityIdentities
  - ▶ krbAlias

# LDAP Kerberos - Representing principals in ACLs

- The bind principal@realm must be represented as a DN for access control.
- Mapping methods
  - ▶ principal@realm mapped directly to the DN  
ibm-kn=principal@realm
  - ▶ principal@realm mapped to its associated SDBM style DNs.
  - ▶ principal@realm mapped to its associated TDBM style DN.
- Mapping principal@realm may yield a list of alternate DNs - Permission will be union of permissions of all alternate DNs

# LDAP Kerberos - Mapping Algorithms

- Assume Kerberos principal "jeff@IBM.COM"
- Direct mapping in ACL  
dn: cn=Scott,o=IBM,c=us  
aclEntry: access-id:ibm-kn=jeff@IBM.COM...
- SDBM (RACF) Mapping  
RACF maps principal@REALM to a RACF userID from Kerberos information in the USER or KERBLINK profiles. It then provides an SDBM distinguished name

racfld=JEFF,profiletype=user,sysplex=plex1



# LDAP Kerberos - Mapping Algorithms

## ■ TDBM (DB2) Mapping

- ▶ Search the entire database for the realm entry.  
krbprincsubtree indicates a list of subtrees where principals can be found

```
dn: krbrealmname-V2=IBM.COM,o=Lotus,c=US
objectclass: krbrealm-V2
krbrealmname-V2: IBM.COM
krbprincsubtree: o=Lotus,c=US
```

- ▶ Look for an entry in the designated subtrees with **KrbPrincipalName**:*jeff@IBM.COM*  
Add the entry's DN to the alternate DN list - Perform group gathering using the list

```
dn: cn=Jeff,o=IBM,c=US
objectclass: extensibleObject
krbPrincipalName:jeff@IBM.COM
```

- ▶ use DN cn=jeff,o=IBM,c=US and associated group(s) for ACL checking

# LDAP Kerberos - Mapping Algorithms

## ■ TDBM other Mapping algorithms

### ► krbAliasedObjectName

dn: cn=Jeff,o=Lotus,c=US

objectclass: krbAlias

objectclass: extensibleobject

krbPrincipalName: jeff@IBM.COM

krbAliasedObjectName: cn=Tim,o=Lotus,c=US

dn: cn=Tim,o=Lotus,c=US

objectClass: krbAlias

krbHintAliases: cn=Jeff,o=Lotus,c=US

Results in cn=jeff,o=Lotus,c=US and cn=Tim,o=Lotus,c=US in the alternate DN list

### ► altSecurityIdentity

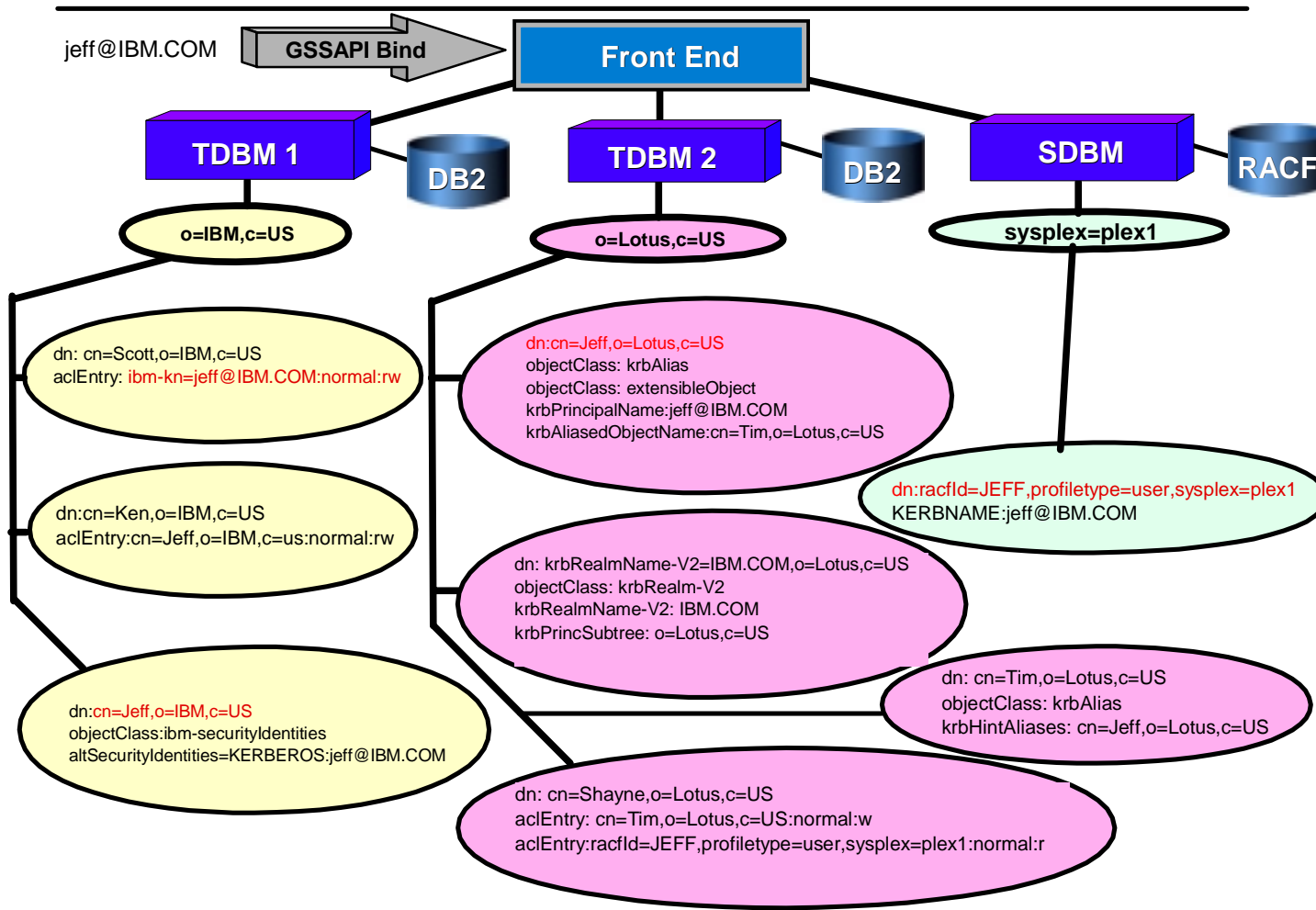
dn: cn=Jeff,o=IBM,c=US

objectclass: ibm-securityIdentities

altSecurityIdentity: KERBEROS:jeff@IBM.COM

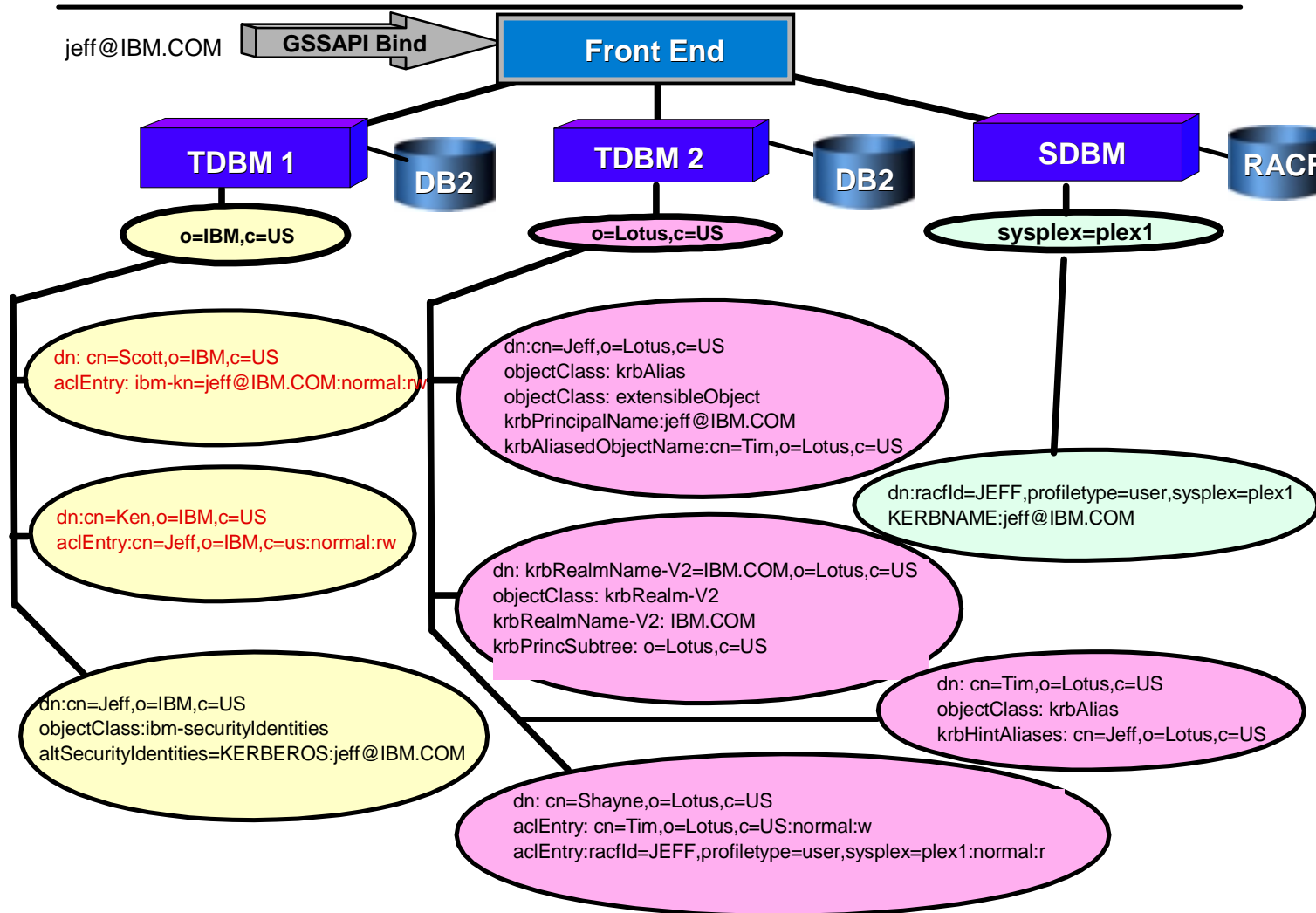
# LDAP Kerberos ACL/Mapping Diagram Results

- The following list of identities will be used for access control:
  - ▶ ibm-kn=jeff@IBM.COM
  - ▶ cn=Jeff,o=IBM,c=US
  - ▶ cn=Jeff,o=Lotus,c=US
  - ▶ racfld=JEFF,profiletype=user,sysplex=plex1



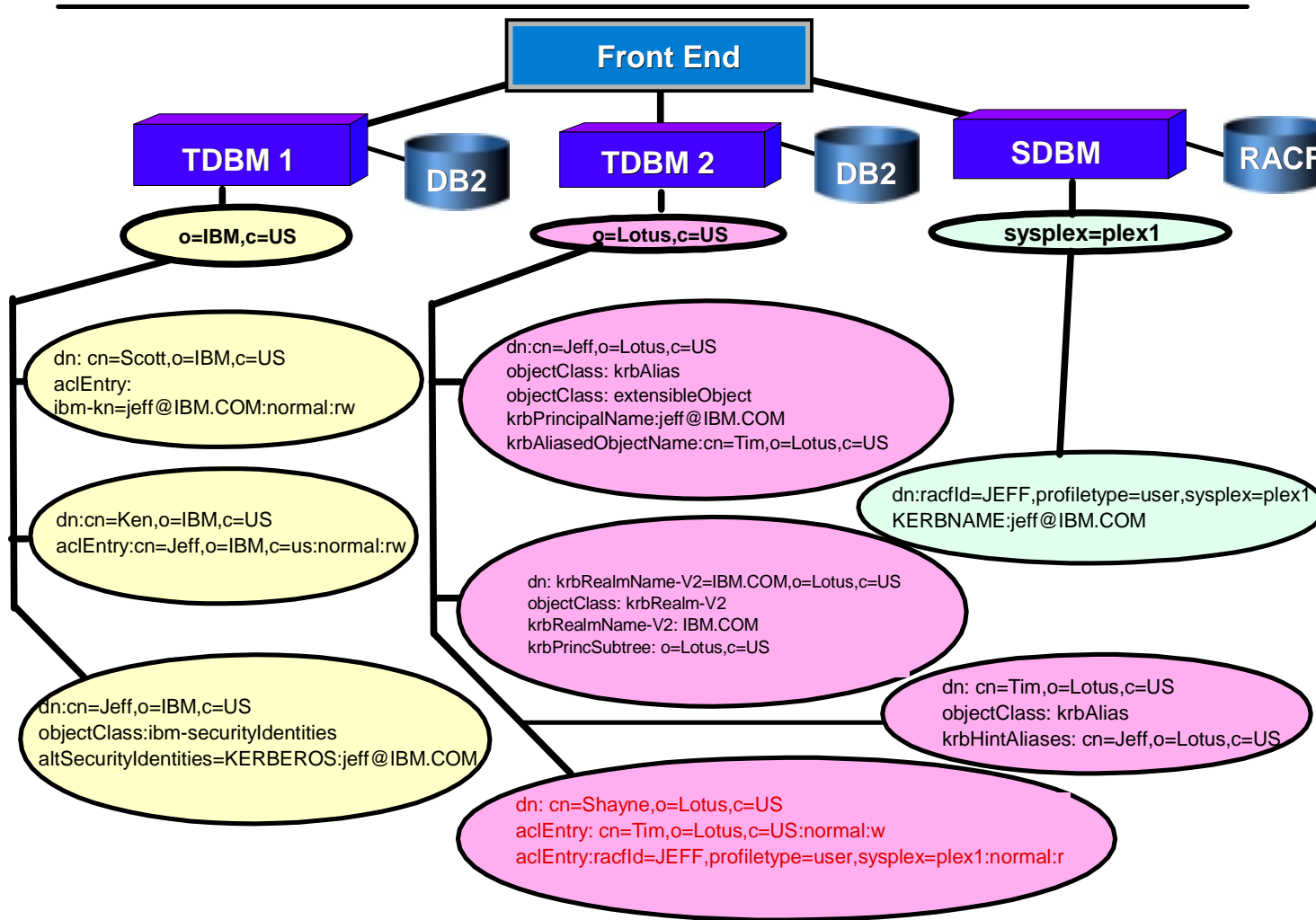
# ACL/Mapping Diagram Results

- jeff@IBM.COM would be able to search and modify the Scott entry since he was mapped to ibm-kn=jeff@IBM.COM and Scott's aclEntry states that he has read and write permission.
- jeff@IBM.COM would also be able to search and modify the Ken entry since he was mapped to cn=Jeff,o=IBM,c=US and Ken's aclEntry states that he has read and write permission.



# ACL/Mapping Diagram Results

- jeff@IBM.COM would also be able to modify the Shayne entry since he was mapped to Tim and Shayne's aclEntry states that Tim has write permission.  
jeff@IBM.COM will also be able to search for the Shayne entry since he mapped to racfld=JEFF,profiletype=user,sysplex=plex1 and Shayne's aclEntry states that JEFF has read permission.



# z/OS 1.2 RACF Group Size Expansion



**Redbooks**

International Technical Support Organization

# RACF Group Size Expansion

- New support that allows the RACF administrator to create groups that may contain more than the current limit of 5957 members
  - unlimited amount of users with USE authority only
  - the 5957 limit still applies to users with more privileges (CONNECT, JOIN, group-SPECIAL/OPERATIONS/AUDITOR)
- UNIVERSAL Keyword on ADDGROUP Command
  - ADDGROUP (GENUSERS) OWNER(GENADMIN) **UNIVERSAL**
- Not available for ALTGROUP
- Access list with universal group in it will work as expected

# RACF Group Size Expansion

- Users are connected as usual
  - CONNECT (GENADMN) GROUP(GENUSERS) AUTHORITY(JOIN)
  - CONNECT (GENUSER1) GROUP(GENUSERS)

## ■ LISTGRP Output

### ▸ LISTGRP GENUSERS

INFORMATION FOR GROUP GENUSRS  
SUPERIOUR GROUP=SYS1 OWNER=GENADMN

....

TERMUACC

**UNIVERSAL**

NO SUBGROUPS

....

USER(S)=	ACCESS=	ACCESS COUNT=	UNIVERSAL ACCESS=
GENADMN	JOIN	000000	NONE

ATTRIBUTES=NONE

REVOKE DATE=NONE

RESUME DATE=NONE



# RACF Group Size Expansion

- How to list ALL users of a UNIVERSAL GROUP
  - ▶ Users actually not in the group profile
  - ▶ Use Database Unload Utility (IRRDBU00) and input to DB2 or RACFICE to query userIDs
  
- Use the Remove ID Utility (IRRRID00) to remove a UNIVERSAL group
  - ▶ You need to execute the REMOVE commands and the DELGROUP command generated by RID

# RACF Group Size Expansion

- R\_admin callable service (IRRSEQ00)
  - Keyword support added for UNIVERSAL keyword on ADMN\_ADD\_GROUP function
- SMF Records
  - Type 6 relocate section for ADDGROUP updated to indicate UNIVERSAL group
- Database Unload (IRRDBU00) Records
  - Type 0100 record for GROUP base segment updated with new field for UNIVERSAL group
- RACF Database Templates
  - Flag field UNVFLG added to base segment of group template

# RACF Group Size Expansion

## ■ COMPATIBILITY & MIGRATION

Down level system sharing data base and executing commands

- ▶ ADDGROUP with UNIVERSAL keyword will fail
- ▶ DELUSER and REMOVE will work as before
- ▶ DELGROUP will not issue any warning messages
- ▶ DELGROUP will succeed if UNIVERSAL groups members all have authority higher than USE and/or have group related user attributes
- ▶ ADDUSER and CONNECT will add users to the UNIVERSAL group, but they WILL BE INCLUDED in 5957 limit
- ▶ ALTUSER, with AUTHORITY(USE) specified WILL INCLUDE the member in the 5957 limit

# RACF Group Size Expansion

## ■ COMPATIBILITY & MIGRATION

- ▶ UP-LEVEL SYSTEM CAN BE USED TO GET AN AUTH(USE) USER NOT COUNTED TOWARD 5957 LIMIT  
If a user IS connected with USE(AUTH) and has no group related user attributes, and IS showing up in LISTGRP output, issuing the CONNECT command with AUTH(USE) will change the member to not be counted towards group member size limit
- ▶ ICH577E WARNING: BASE SEGMENT OF GROUP TEMPLATE AT LEVEL XXXXXX DOES NOT CONTAIN FIELD UNVFLG  
Issued if system has UNIVERSAL support but templates are down level

# z/OS 1.2 SAFTRACE Facility



**Redbooks**

International Technical Support Organization

# SAFTRACE

- Provides tracing of RACROUTE, SAF callable service, and ICHEINTY requests to aid problem diagnosis
- Available only at z/OS 1.2 and above
- Enabled via RACF subsystem SET TRACE command
- Can specify which requests to trace and which address spaces to trace
  - SET TRACE( JOBNAME(xyz) RACROUTE( TYPE(1) ))  
will trace all RACROUTE REQUEST=AUTH from job xyz
  - SET TRACE( ASID(25) DATABASE(ALTER) )  
will trace all ICHEINTY ALTER, ADD, DELETE, RENAME from address space 25
- Trace goes to GTF, like other RACF SET TRACE output
- Use IPCS to read the trace, with the GTF USRcommand

# SAFTRACE

- Trace points

- IBM SAF routers ICHSFR00 and IRRSFR11

- Internal calls to the security product may not be traced.

- All calls made via RACROUTE or Callable Service interface will be traced.

- Calls that issue SVC (pre-RACROUTE) or directly enter the security product will not be traced.

- RACF Database manager ICHEINTY interface

- All ICHEINTYs and internal security product calls to the database manager.

# SAFTRACE

- Situations where SAFTRACE can be helpful:
  - If RACF database contention has been observed:

Trace DATABASE(ALTER) requests on the specific ASID indicated via GRS contention displays. Alter requests generally prevent readers (majority) from getting service
  - Excessive database i/o (for a given address space)

Trace reads to see what CLASS / ENTITIES are related
  - Excessive Verify's:

If your systems has an excessive amount of Verify's, set a trace on RACROUTE(TYPE(2,5,9)) and determine who is issuing all of the RACROUTE calls.



# SAFTRACE

```
[subsystem-prefix] SET [TRACE(  
    [ RACROUTE(ALL | NONE | TYPE(t1, t2,...)) |  
      NORACROUTE ] |  
  
    [ DATABASE ([ALL | NONE] |  
        [ALTER | NOALTER] |  
        [ALTERI | NOALTERI] |  
        [READ | NOREAD]) |  
      NODATABASE ] |  
  
    [ CALLABLE(ALL | NONE | TYPE(t1, t2, ...)) |  
      NOCALLABLE]  
  
    [ ASID(asid1, asid2, .. | *.) | NOASID | ALLASIDS ]  
    [ JOBNAME(jobname1, jobname2, ... | *) |  
      NOJOBNAME | ALLJOBNAMES ]  
    )]
```

See the services type numbers in the appendices

# SAFTRACE

1. Start the GTF trace  
**START GTFRACF.GTF,,,NOPROMPT**  
(see sample procedure on next foil)
2. Use the SET command to enable your trace:  
**@SET TRACE(CALLABLE(37))**  
**JOBNAME(J23DC002))**
3. Reproduce the scenario that trace is required for
4. Next stop GTF to prevent excessive traces  
**STOP GTF**
5. Use IPCS to view the trace data.

The input trace data is contained in the dataset specified on the IEFRDER DD card in the GTFRACF (or other) procedure.  
Once the TSO IPCS session is active use the IPCS subcommand "IP GTF USR" to display the formatted trace

# SAFTRACE

```
//GTFRACF PROC MEMBER=GTFRM#O
//BR14 EXEC PGM=IEFBR14,REGION=512K
//SYSPRINT DD SYSOUT=*
//D DD DISP=(OLD,DELETE),UNIT=3380,VOL=SER=TEMP01,
// DSN=SYS1.TRACE
//IEFPROC EXEC
PGM=AHLGTF,PARM='MODE=EXT,DEBUG=NO,SA=100K,AB=100K',
// REGION=2880K,TIME=NOLIMIT
//IEFRDER DD
DSNAME=SYS1.TRACE,UNIT=3380,VOL=SER=TEMP01,
// DISP=(NEW,CATLG),SPACE=(TRK,(100))
//SYSLIB DD
DSNAME=RACFDRVR.PARMLIB.R6(&MEMBER),DISP=SHR
```

Sample Parmlib Member: GTFRM#O

TRACE=USRP

USR=(F44),END

# SAFTRACE

## Usage notes

Things to know:

- The RACF subsystem must be up and running
- GTF must be active
- Trace information is not saved across IPLs
- For OMVS calls, you need and '\*' in the jobname filter to trace spawned processes. Otherwise, you will not get a complete set of records. Example:  
`SET TRACE(CALLABLE(ALL) JOBNAME(IBMUSER*))`  
will trace on jobnames IBMUSER1, IBMUSER2, etc..

# SAFTRACE

Sample SET LIST output after issuing the following command:  
SET TRACE(CALLABLE(TYPE(2,5,9)) JOBNAME(IBMUSER\*))

```
- RACFR12  IRRH005I (@) RACF SUBSYSTEM INFORMATION:
-   TRACE OPTIONS                                - NOIMAGE
-                                                    - NOAPPC
-                                                    - RACROUTE
-                                                    2  5  9
-                                                    - NOCALLABLE
-                                                    - NODATABASE
-                                                    - NOASID
-                                                    - JOBNAME
-                                                    IBMUSER*
-   SUBSYSTEM USERID                            - IBMUSER
-   JESNODE (FOR TRANSMITS)                     - POKVMMCL
-   AUTOMATIC COMMAND DIRECTION IS *NOT* ALLOWED
-   AUTOMATIC PASSWORD DIRECTION IS *NOT* ALLOWED
-   PASSWORD SYNCHRONIZATION IS *NOT* ALLOWED
-   AUTOMATIC DIRECTION OF APPLICATION UPDATES IS *NOT* ALLOWED
-   RACF STATUS INFORMATION:
-       TEMPLATE VERSION                        - HRF7705
00-       DYNAMIC PARSE VERSION                  - HRF7705
```

# SAFTRACE

## Output trace format

Header  
information  
(fixed length)

Unloaded parameters  
from RACF  
parameter list

Raw hex dump  
of entire GTF  
record including  
header

```
Trace Identifier:      00000036
Record Eyecatcher:    RTRACE
Trace Type:           RACFPRE
Ending Sequence:      .....
Calling address:       00000000  8B04A24E
Requestor/Subsystem:  RSSC06    RACF
Task address:         00000000  006EC1A0
Task ACEEP:          00000000  00000000
Time:                 B5773AAD  0E780C4B
Error class:          .....
Service number:       00000005
RACF Return code:     00000000
RACF Reason code:     00000000
Return area address:   00000000  00000001
Parameter count:      0000000A
```

```
Area length:          00000068
```

```
Area value:
```

```
00000000 00000000 00680200 00055800 |
```

```
.....|
0B089158 0B089160 0B08916C 00000000 |
```

```
.j...j-..j%....|
```

```
00000000 00000068 00000000 00000000 |
```

```
.....|
```

```
00400000 00000000 00000000 00000000 | .
```

```
.....|
```

```
00000000 00000000 00000000 00000000 |
```

```
.....|
```

```
00000000 00000000 00000000 00000000 |
```

```
.....|
```

```
00000000 00000000 | .....
```

```
Area length:          0000006C
```

```
Area value:
```

```
6C0000A0 00000000 00000000 00000000 |
```

```
%.....|
```

```
00000000 00000000 00000000 00000000 |
```

```
.....|
```

```
00000000 00000000 00000000 00000000 |
```

```
.....|
```

# SAFTRACE

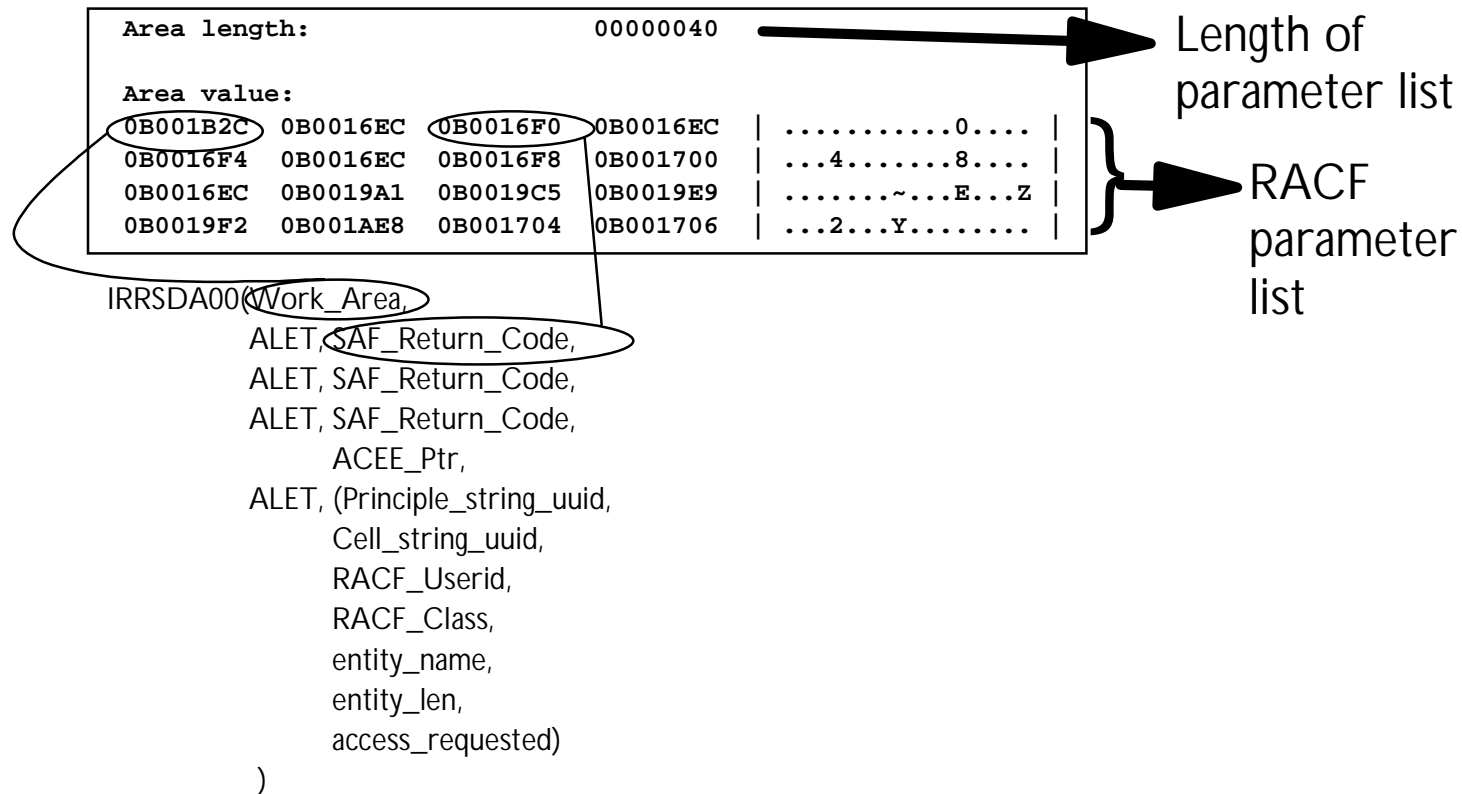
## To read a callable services trace

Trace Identifier:	00000036	
Record Eyecatcher:	RTRACE	
Trace Type:	OMVSPRE	
Ending Sequence:	.....	
Calling address:	00000000	8B000BAE
Requestor/Subsystem:	.....	.....
Primary jobname:	J23DC002	
Primary asid:	0000012E	
Primary ACEEP:	00000000	006F6700
Home jobname:	J23DC002	
Home asid:	0000012E	
Home ACEEP:	00000000	006F6700
Task address:	00000000	006F6B00
Task ACEEP:	00000000	00000000
Time:	B570BA31	042B3327
Error class:	.....	
Service number:	00000025	← R_dceauth
RACF Return code:	00000000	
RACF Reason code:	00000000	
Return area address:	00000000	00000000
Parameter count:	00000025	

# SAFTRACE

## To read a callable services trace

The first parameter on a callable service is the RACF parameter list.





# SAFTRACE

## To read a callable services trace

The rest of the parameters are unloaded out of the parameter list.

Area length:	00000008	}	"EYECATCHER"
Area value:			
D6C6C6E2 C5E30004	OFFSET..		
Area length:	00000004	}	First ALET value
Area value:			
00000000	....		
Area length:	00000008	}	"EYECATCHER"
Area value:			
D6C6C6E2 C5E30008	OFFSET..		
Area length:	00000004	}	SAF Return Code value
Area value:			
00000000	....		

# SAFTRACE

## To read a callable services trace

```
Area length:          00000008
Area value:
D6C6C6E2  C5E3001C      | OFFSET..      |
Area length:          00000004
Area value:
7FFF5268      | "...      |
Area length:          000000A8
Area value:
C1C3C5C5  FF0000A8  02000000  00000000 | ACEE...Y..... |
00000000  05C8D6D5  C4C14040  4004E2E8 | .....HONDA  .SY |
.....
00000000  00000000  7FFD3290  00000000 | ....."...... |
00000000  7FFD3968      | ....".... |
Area length:          00000050
Area value:
50010000  0000C000  00000000  00000000 | &.....{..... |
.....
C8D6D5C4  C1404040  E2E8E2F1  40404040 | HONDA  SYS1 |
Area length:          00000090
Area value:
C1C3C5E7  03000000  00000000  00000000 | ACEX..... |
00000000  00000000  00000000  00000000 | ..... |
.....
```

} "EYECATCHER"

} ACEPTR from  
parameter list

This is the actual ACEE  
along with the TOKEN  
and ACEX.

Note: there are no "eyecatchers"  
since it's essentially all one  
parameter.

# SAFTRACE

To read a SAFTRACE - Special control blocks and other handling

- ACEE: not only is the ACEE unloaded, but so are the USP, TOKEN, and ACEX if available.
- CRED: (IRRPCRED) After the CRED structure is unloaded, the first path name, the second path name, the first filename and second filename are unloaded.
- Work Areas: Work Areas in general are not unloaded.
- Passwords: Are not unloaded.
- Installation parameters
- ENVIRIN and ENVIROUT parameters
  - Certificates

# SAFTRACE

## APPENDIX

RACROUTE REQUEST=	Service Number or TYPE (HEX)	Service Number or Type (Decimal)
AUTH	1	1
FASTAUTH	2	2
LIST	3	3
DEFINE	4	4
VERIFY	5	5
EXTRACT	6	6
DIRAUTH	7	7
TOKENMAP	8	8
VERIFYX	9	9
TOKENXTR	A	10
TOEKNBLD	B	11
EXTRACT, BR=YES	C	12
AUDIT	D	13
STAT	E	14
SIGNON	F	15
TOKENMAP, XMEM	10	16
TOKENXTR, XMEM	11	17

CALLABLE SERVICE	Service Number or TYPE (HEX)	Service Number or TYPE (DECIMAL)
IRRRIU00 - initUSP	1	1
IRRRDU00 - deleteUSP	2	2
IRRRMF00 - makeFSP	3	3
reserved	4	4
IRRRMM00 - R_umask	5	5
IRRRKA00 - ck_access	6	6
IRRRKP00 - ck_priv	7	7
IRRRUM00 - getUMAP	8	8
IRRRGM00 - getGMAP	9	9
IRRRGG00 - R_getgroups	A	10
IRRRSU00 - R_setuid	B	11
IRRRU00 - R_seteuid	C	12
IRRRSG00 - R_setgid	D	13
IRRRREG00 - R_setegid	E	14
IRRRCO00 - R_chown	F	15

# SAFTRACE

## APPENDIX

CALLABLE SERVICE	Service Number or TYPE (HEX)	Service Number or TYPE (DECIMAL)
IRRRCF00 - R_chmod	10	16
IRRRC A00 - R_chaudit	11	17
IRRREX00 - R_exec	12	18
IRRRAU00 - R_audit	13	19
IRR RK000 - ck_process_owner	14	20
IRR RQS00 - query_system_security_options	15	21
IRR RQF00 - query_file_security_options	16	22
IRR RCS00 - clear_setid	17	23
IRR RK F00 - ch_file_owner	18	24
IRR RM R00 - make_root_FSP	19	25
IRR RPT00 - R_ptrace	1A	26
IRR RUG00 - R_getgroupsbyname	1B	27
IRR RFK00 - R_fork	1C	28
IRR RMI00 - makeISP	1D	29
IRR RKI00 - ck_IPC_access	1E	30

CALLABLE SERVICE	Service Number or TYPE (HEX)	Service Number or TYPE (DECIMAL)
IRRRCI00 - R_IPC_ctl	1F	31
IRRRC200 - ck_owner_two_files	20	32
IRR RGE00 - get_uid_gid_supgrps	21	33
IRR RDI00 - R_dceinfo	22	34
IRR RDK00 - R_dcekey	23	35
IRR RUD00 - R_dceruid	24	36
IRR RDA00 - R_dceauth	25	37
IRRRIA00 - Initacee	26	38
*IRRSEQ00 - R_admin	27	39
*IRR SIM00 - R_usermap	28	40
*IRRSDL00 - R_datalib	29	41
*IRRSMK00 -	2A	42
*IRRSPK00 - R_ticketserve	2B	43
IRRSPX00 - R_PKIServ	2C	44
IRR SCH00 - R_cacheserv	2D	45
IRRSPY00 - R_proxyserv	2E	46

# **z/OS 1.2 RACF Support of Mixed Cases Profiles Names**



**Redbooks**

International Technical Support Organization

# RACF Support of Mixed Cases Profile Names - Objectives

- Support Enterprise Java Bean (EJB) authorization roles in RACF profiles
  - ▶ For WebSphere and CICS
  - ▶ For EJB compliance test suite
  - ▶ EJB roles must allow mixed case
- Benefit: Customers/applications can now allow use of mixed case profile names in customer-defined RACF classes
- Available now with APAR OW46859 on
  - ▶ OS/390 V1R8 with PTF UW78360
  - ▶ OS/390 V1R10 with PTF UW78361

# RACF Support of Mixed Cases Profile Names - Changes to Externals

- Add new CASE=UPPER|ASIS keyword to ICHERCDE macro
  - ▶ ICHERDCE defines a class entry in the CDT
- Add two new RACF classes
  - ▶ EJBROLE and GEJBROLE (member and grouping class pair)
  - ▶ Specify CASE=ASIS for these classes
- Update RACF commands to accept mixed case text where appropriate
- Update the appropriate ISPF panels



# RACF Support of Mixed Cases Profile Names - Enhanced RACF Commands

- RDEFINE, RALTER, RLIST, RDELETE, and PERMIT accept mixed-case profile names for CASE=ASIS classes
- RDEFINE, PERMIT, and ADDSD accept mixed-case profile names in the FROM keyword when FCLASS (or base class if FCLASS omitted) is a CASE=ASIS class
- RDEFINE accepts mixed-case member names on ADDMEM keyword for CASE=ASIS classes
- RALTER accepts mixed-case member names on ADDMEM and DELMEM keywords for CASE=ASIS classes

# RACF Support of Mixed Cases Profile Names - Enhanced RACF Commands

- SEARCH command accepts mixed-case MASK and FILTER strings for CASE=ASIS classes
- SEARCH command accepts mixed-case CLIST strings (regardless of class) and places "CONTROL ASIS" statement in output CLIST when at least one CLIST string is specified (presumably, commands are being generated)
  - ▶ Allows commands to be issued against mixed-case profile names without manually changing CLIST
  - ▶ Fixes existing limitation where certain commands won't work:
    - **SEARCH CLASS(USER) CLIST('ALTUSER ' 'OMVS(PROGRAM(/bin/sh))' )**
  - ▶ Omitting "CONTROL ASIS" when CLIST string not specified reduces interference with data files intended for use by programs

# RACF Support of Mixed Cases Profile Names - Migration Considerations

- No existing RACF classes were changed to CASE=ASIS, and customer changes to RACF classes are not supported
- Customers can use CASE=ASIS for their own new or existing classes
  - ▶ Must keep CDTs in synch across nodes when using RACF Remote Sharing Facility (RRSF). This is business as usual.
  - ▶ When changing an existing class to CASE=ASIS, they must educate their users, because RACF will accept profile names in the case in which they are typed
    - **If users are used to typing in lower case and relying on RACF to upper-case profile names, then RACF will be creating profiles which they do not expect!**