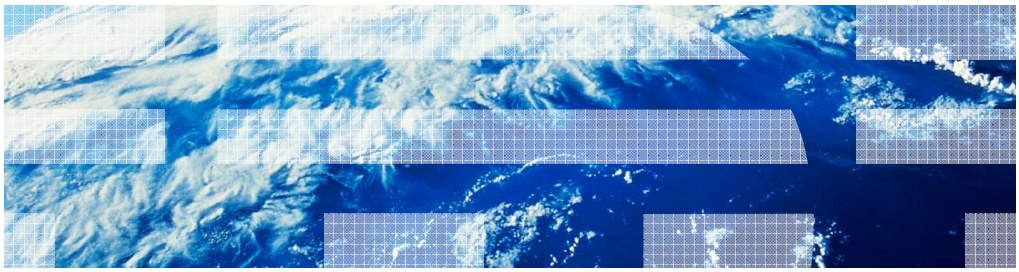
# z/OS V1R13

## IBM Tivoli Directory Server and LDAP: z/OS V1R13 enhancements



## Session objectives

- Explain the purpose of each IBM TDS for z/OS® R13 line item:
  - 64-bit DB2®-based backends
  - Server support for Paged and sorted search results
  - Group search limits
  - SHA-2 and Salted SHA-2 hashing
  - Kerberos client updates and INADDR_ANY/in6addr_any listen option support
  - Administrative group and roles
- Identify new/changed installation procedures of each TDS line item
- Define the functional content and benefit of each TDS line item
- Explain any migration issues or concerns of each TDS line item
- Explain how the function of each TDS line item is invoked
- Indicate list of Publications and References

Section

# 64-bit DB2-based backend support

## Overview

- Problem Statement / Need Addressed
  - Only 31-bit addressing mode supported in TDBM and DB2-based GDBM backends
  - Not enough scalability when large directories and/or large entries need to be handled
- Solution
  - Add support for 64-bit addressing mode for:
    - TDBM and DB2-based GDBM backends
    - ldif2ds (bulkload) utility
    - ds2ldif (unload) utility
- Benefit / Value
  - Can access the address space above the 2GB bar
  - Reduce memory constraint issues when using large directories or entries in the server




In the past, TDBM does not support 64-bit addressing mode (AMODE). With 31-bit AMODE, the maximum address space that applications can allocate is approximately 2G.

This line item enhances z/OS LDAP to support 64-bit addressing mode in the TDBM backend and the bulkload (**ldif2ds**) utility, unloading (**ds2ldif**) from 64-bit a TDBM backend, and using 64-bit DB2-based GDBM.

## Usage & Invocation

- Configuration file updates:
  - Must specify the 64-bit DLL name for TDBM or GDBM on the **database** option
  - For example:
    - database tdbm GLDBTD31/**GLDBTD64**
    - database gdbm GLDBGD31/**GLDBGD64**
  - Must update the LDAP server's procedure to run GLDSRV64 instead of GLDSRV31
- **dsconfig** utility
  - Can be used to configure server with 64-bit TDBM and/or DB2-based GDBM
- The **ds2ldif** (unload) and **ldif2ds** (bulkload) utilities have been updated to support TDBM in 64-bit addressing mode

## Interactions & Dependencies

- Software Dependencies
  - One of the following is required:
    - DB2 version 9 with PTF UK50918 (DB2 ODBC 64-bit support)
    - DB2 version 10 or higher
  - DB2 version 9 PTF UK55577 (APAR PM05254)
- Hardware Dependencies
  - None
- Exploiters
  - None




PTF 55577 (APAR PM05254) is only a fix pack that resolves a bug in the DB2 64-bit ODBC driver.

## Migration & Coexistence Considerations

- Migration:
  - None
- Coexistence:
  - If sharing a DB2-based backend with a server running in 31-bit addressing mode, the 31-bit server may have memory constraint problems handling the large number of entries or the size of the entries that can be created by the 64-bit server

## Installation

- New modules/programs in SYS1.SIEALNKE:
  - GLDUTB64 - 64-bit TDBM/GDBM utility module
  - GLDBTD64 - 64-bit TDBM backend module
  - GLDBKL64 - 64-bit **ldif2ds** (bulkload) utility
- Updated modules/programs in SYS1.SIEALNKE:
  - GLDBGD64 – Updated to support running DB2-based GDBM backends in 64-bit mode
  - GLDUNL64 – Updated to support unloading TDBM backends in 64-bit mode (**ds2ldif** utility)

## Installation

- New sample files for running the ldif2ds utility in 31-bit or 64-bit addressing mode
  - Shell scripts
    - ldif2ds31 (same as ldif2ds) and ldif2ds64
    - /usr/lpp/ldap/sbin
  - JCL
    - LDF2DS31 (same as LDF2DS) and LDF2DS64
    - *GLDHLQ*.SGLDSAMP
  - REXX
    - LDF2DS31 (same as LDF2DS) and LDF2DS64
    - *GLDHLQ*.SGLDEXEC

Section

# *Server support for paged and sorted search results*

## Overview (1 of 2)

- Problem Statement / Need Addressed
  - TDS for z/OS does not have the ability to page or sort search results
  - Distributed TDS does have this ability
- Solution
  - TDS for z/OS is providing server-side paging and sorting of search results based on IETF RFCs 2696 and 2891
- Benefit / Value
  - Customers who exploit distributed TDS paged and sorted support will be able to exploit the same support on z/OS
  - LDAP clients without paging or sorting capabilities can retrieve paged and sorted search results




TDS - Tivoli® Directory Server

This support is being provided in z/OS V1.13 for the server and the client ldapsearch utility. Client API support is already available.

The following RFCs are implemented by this line item:

•RFC 2696 LDAP Control Extension for Simple Paged Results Manipulation.

•RFC 2891 LDAP Control Extension for Server Side Sorting of Search Results

## Overview (2 of 2)

- Paged and sorted search results can be requested on an LDAP search:
  - Paged search results provides paging capabilities for LDAP clients that want to receive just a subset of search results at a time.
  - Sorted search results enables an LDAP client to receive sorted search results based on a list of criteria, where each criterion represents a sort key
    (example: sort by cn attribute).

## Usage & Invocation (1 of 4)

- Configuration file updates
  - **idleConnectionTimeout** (updated)
    - Number of seconds the LDAP server will wait on idle connection or idle paged search result set (default=0).
- cn=configuration entry – new attributes
  - **ibm-slapdPagedResLmt**
    - Maximum number of outstanding paged search requests allowed simultaneously on a single connection (default=0)
  - **ibm-slapdPagedResAllowNonAdmin**
    - Indicates whether the server allows non-administrators to request paged search results (default=false).
  - **ibm-slapdSortKeyLimit**
    - Maximum number of sort keys that can be included on a single sorted search request (default=0).




Note that paged and sorted search results are disabled by default.

The **ibm-slapdPagedResLmt** attribute must be set to a value greater than zero to enable paged search results. The **ibm-slapdPagedResAllowNonAdmin** attribute can be set to true to enable paged search requests from non-administrators. Additionally, the **idleConnectionTimeout** configuration option will determine the length of time before the LDAP server's network monitor task abandons idle paged results.

The **ibm-slapdSortKeyLimit** attribute must be set to a value greater than zero to enable sorted search results. The **ibm-slapdSortSrchAllowNonAdmin** attribute can be set to true to enable sorted search requests from non-administrators.

## Usage & Invocation (2 of 4)

- **ibm-slapdSortSrchAllowNonAdmin -** Indicate whether the server allows non-administrators to request sorted search results (default=false)

- New server control support:
  - **pagedResults -** Used on a SearchRequest and SearchResultDone message to control the rate at which the server returns search results
  - **SortKeyRequest -** Used on a SearchRequest message to specify the criteria that a server should use to sort the results of an LDAP search request
  - **SortKeyResponse -** Used on a SearchResultDone message to return the result of a sorted search.




The **ibm-slapdSortSrchAllowNonAdmin** attribute can be set to true to enable sorted search requests from non-administators.

The new cn=configuration attributes are added to the minimum schema and schema.IBM.ldif.

Like distributed TDS, we are defining an administrator bind as a bind from the LDAP administrator or any user with a defined administration role. This means that when sorted or paged search requests are limited to administrators, anyone defined with an administrator role is allowed to make these requests.

## Usage & Invocation (3 of 4)

- **ldapsearch** client utility is updated with new options
  - **-o** sortKey
    - Specifies a sort key that the server should order search results by
  - **-q** pageSize
    - Specifies a page size and requests that the server return search results in pages with the number of entries matching the page size
  - **-T** pageTime
    - Specifies the number of seconds between paged search requests




**-o** *sortKey*

Specifies a sort key that the server should order search results by. Multiple -o options can be specified to further define the sort order. An optional minus sign ( - ) specified as a prefix on the sortKey indicates to sort the results in reverse order. This option directs the utility to send and receive the sorted search request and response controls. The criticality of the request control is always critical.

**-q** *pageSize*

Specifies a page size and requests that the server return search results in pages with the number of entries matching the page size. Multiple **-q** values can be specified to request pages of different sizes. In this case, the first **-q** value is used on the first page request, the second **-q** value is used on the second page request, and so on. If there are more pages than **-q** values, the last **-q** value is used for all remaining pages. The last page returned may contain fewer entries than the requested **-q** value. This option directs the utility to send and receive the paged search result control. The criticality is always critical.

**-T** *pageTime*

Specifies the number of seconds between paged search requests. This option requires the **-q** option. An alternative to the **-T** option for requesting a subsequent page is to press the Enter key after paged results are returned.

## Usage & Invocation (4 of 4)

- **ldapsearch** example:
  ldapsearch -b "o=Deltawing, c=AU" -o sn -o -ibm-slapdDN -q 3 -q 2 -T 10 -v "(|(sn=Harris)(sn=Stephens))" sn
  - This command performs a verbose sorted and paged search, sorting first by SN, then by DN (descending), with each subsequent page requested 10 seconds after the preceding page is returned.
  - Assume a total of 8 entries are returned from this search. The first page contains 3 entries, the second and third page each contain 2 entries, and the last page contains the final entry.

## Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - Client applications which issue search requests or users of the ldapsearch utility who:
    - retrieve large amounts of directory data, or
    - retrieve LDAP directory data in an organized manner.

## Migration & Coexistence Considerations

- Migration
  - None
- Coexistence
  - None

## Installation

- By default this support is not activated.
  It is activated by updating the cn=configuration entry as mentioned a few slides ago.

# *Group search limits*

## Overview (1 of 2)

- Problem Statement / Need Addressed
  - Some users need larger search limits, but raising the limits for all users is unacceptable
    - System capacity or Company policy
- Solution
  - Associate search limits with an LDAP group
- Benefit / Value
  - As many group search limits as needed
  - Can change without restarting the server
  - Compatible with Distributed TDS




As a security precaution company policy may constrain the amount of data retrievable by the general user.

## Overview (2 of 2)

- To define group search limits, add the **ibm-searchLimits** auxiliary objectclass to the group entry
  - Three required attribute types: cn, ibm-searchTimeLimit, and ibm-searchSizeLimit
  - If user belongs to multiple groups, the group with the largest size or time limit takes precedence (unless it is 0 or -1)
- **ibm-searchTimeLimit** - Maximum number of entries to return from search requests for a member in a special search limit group. 0 = unlimited. -1 = ignored.
- **ibm-searchSizeLimit** - Maximum number of seconds to spend on search requests for a member in a special search limit group. 0 = unlimited. -1 = ignored.

## Usage & Invocation (1 of 2)

- Example 1:
  - Server configured size limit of 10 and time limit of 30 seconds.
  - User requests 20 entries.
  - 10 entries returned.
- Example 2
  - Same server limits.
  - User is member of group with size limit of 100 and time limit of 60 seconds.
  - Same request.
    - 20 entries returned.

## Usage & Invocation (2 of 2)

- Example 3
  - Server configured size limit of 10 and time limit of 30 seconds.
  - User is member of group with size limit of 100 and time limit of 60 seconds.
  - User requests all entries (0=unlimited).
  - 100 entries returned.
- Example 4
  - Same server limits.
  - User is an administrator.
  - User requests all entries (0=unlimited).
    - All entries returned.

## Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

## Migration & Coexistence Considerations

- Migration
  - None
- Coexistence
  - Must be at server compatibility level 7 or greater to use this feature

## Installation

- None

Section

# *SHA-2 and salted SHA-2 hashing*

## Overview (1 of 2)

- Problem Statement / Need Addressed
  - Need to support SHA-2 hash algorithms as NIST government directive indicates that SHA-1 hashing is no longer supported in government applications
- Solution
  - Add SHA-2 and Salted SHA-2 hash algorithms for password hashing
    - SHA-2 includes: SHA224, SHA256, SHA384, and SHA512
- Benefit / Value
  - More secure password hashing algorithms are now provided
  - Better compatibility with other LDAP servers




As of z/OS V1R12, the z/OS LDAP server supports encrypting userpassword attribute values in the LDBM, TDBM, and CDBM backends in AES, DES, md5, crypt, SHA, and SSHA algorithms. The md5, crypt, SHA, SSHA algorithms are one-way hash algorithms, while AES and DES are two-way encryption methods. An NIST government directive indicates that government applications should no longer use the SHA-1 hashing algorithm by the end of 2010. In TDS, the SHA-1 password encryption methods are SHA and Salted SHA (SSHA).

Because of this directive, IBM TDS is adding SHA-2 password hashing support to the server. SHA-2 consists of the following hash algorithms: SHA224, SHA256, SHA384, and SHA512. In addition, IBM TDS will introduce salted versions of each of these algorithms thereby increasing the total to 8 new encryption methods.

Given the same input, a one-way hash algorithm always generates the same resulting hash output. So to help vary the resulting password hash, salted versions of each of these hash algorithms will be introduced. Therefore, this will result in 8 total new encryption methods being added.

Other LDAP servers such as openDS, openLDAP, and non-z/OS IBM TDS 6.3 are or will soon be supporting SHA-2 algorithms. The openDS and non-z/OS IBM TDS 6.3 servers are or will be supporting Salted SHA-2 algorithms.

## Overview (2 of 2)

- Updated **pwEncryption** configuration option to accept **SHA224, SSHA224, SHA256, SSHA256, SHA384, SSHA384, SHA512,** and **SSHA512** as a valid option value
- **ds2ldif** and **ldif2ds** utilities enhanced to handle SHA-2 and Salted SHA-2 algorithms
- Hash new **ibm-slapdAdminPw** attribute according to the **pwEncryption** setting
- Server compatibility level must be 7
  - To set **pwEncryption** option to SHA-2 or Salted SHA-2
  - To accept tagged SHA-2 password values




The **pwEncryption** configuration option indicates the current password encryption or hashing algorithm in use in the LDBM, TDBM, or CDBM backend. In z/OS V1R12, the **pwEncryption** configuration option has been updated to support the following 8 new values for SHA-2 and Salted SHA-2: **SHA224, SSHA224, SHA256, SSHA256, SHA384, SSHA384, SHA512,** and **SSHA512**.

The **ds2ldif** (unload) utility has been updated to support the unloading of SHA-2 and Salted SHA-2 encrypted **userPassword** values. Also, the **ldif2ds** (bulkload) utility has been updated to support encrypting or hashing unencrypted **userPassword** values in SHA-2 and Salted SHA-2 when the **pwEncryption** option is set to the appropriate level.

The server must be running at compatibility level 7 to use SHA-2 and Salted SHA-2 hashing.

## Usage & Invocation (1 of 2)

- SHA-2 and Salted SHA-2 hashing examples
- dn: cn=sha224,o=uid
- objectclass: person
- sn: sha224
- userpassword: {SHA224}lcf7ypKsUIOv2mKlZKPQFPw7cskUDjy5nqa/Eg==

- dn: cn=ssha224,o=uid
- objectclass: person
- sn: sha224
- userpassword: {SSHA224}sEHZjTzABWiPMDSFmnuYDUUMzGF1C+XOcryro0otC3X3smmwNlAbs+
- 222PJREIE7GJUZ4adtbpo=

- dn: cn=sha256,o=uid
- objectclass: person
- sn: sha256
- userpassword: {SHA256}K7gNU3sdo+OL0wNhqoVWhr3g6s1xYv72ol/pe/Unols=

- dn: cn=ssha256,o=uid
- objectclass: person
- sn: ssha256
- userpassword: {SSHA256}qFzEm0vg2BtJL0cK6baEv6VrRJ4MI+wqQtvoknWjEA5lAL3ePW2u0l
- Ur6q+Ye/UYJG+eOyaAuHEehFN3OkGJwA==




The underlying password values for each of these entries is “secret”.

For the salted versions of the SHA-2 algorithms, both us and distributed TDS will be using salt lengths equal to the resulting hash length. These salt values help to introduce some randomness into the resulting hash.

SSHA224 – 28 bytes

SSHA256 – 32 bytes

SSHA384 – 48 bytes

SSHA512 – 64 bytes

## Usage & Invocation (2 of 2)

- dn: cn=sha384,o=uid
- objectclass: person
- sn: sha384
- userpassword: {SHA384}WKd1ukESvjAFrkQHznV9iP2nHUBJe7gCbsrFTU4//HIyzo3jq1rLMK4
- 5dg/ufFPt

- dn: cn=ssha384,o=uid
- objectclass: person
- sn: ssha384
- userpassword: {SSHA384}mbP0pQkuXYlDswEDq6JYWp2Y95jgysAX0wohTmbKP74tQvnkRl9G5e
- u46qth1jOKfvm7HItIuCzcdSRMTe80vynEsv+l0eSfge6Ou3yrXs0cNeN/yw5yMp+FUX0HIg4f

- dn: cn=sha512,o=uid
- objectclass: person
- sn: sha512
- userpassword: {SHA512}vSsar3708Jvp9Szi2NWZZ02Bqp1qRCFpbcTZPdBhnWgs5WtNZKnvCXd
- hztmeD2cmW192CF5bDufKRpayrW/isg==

- dn: cn=ssha512,o=uid
- objectclass: person
- sn: ssha512
- userpassword: {SSHA512}rR/ls84oX0qz/GuxGsdtKaRwhdBXDVEP3Uj/WIRB+KB7zON8DX48gA
- L1k1QCRnrLv0jvyBEB45Dmyj71Awt3M2T5PeagtoTlxbDs1XgVH7zDqAHosWJEI0ZnOviQFP3Cx6
- lR3OMOtd5XEAJKC3RBTnhYkOXmdqqwe6KkorUdaMQ=




The underlying password values for each of these entries is “secret”.

For the salted versions of the SHA-2 algorithms, both us and distributed TDS will be using salt lengths equal to the resulting hash length. These salt values help to introduce some randomness into the resulting hash.

SSHA224 – 28 bytes

SSHA256 – 32 bytes

SSHA384 – 48 bytes

SSHA512 – 64 bytes

## Interactions & Dependencies

- Software Dependencies
  - Must issue these RACF® commands to grant LDAP server's userID access to ICSF SHA-2 hashing routine

```
RDEFINE CSFSERV CSFOWH UACC(NONE)
PERMIT CSFOWH CLASS(CSFSERV) ID(LDAPSRV) ACCESS(READ)
SETROPTS RACLIST(CSFSERV) REFRESH
```

- Hardware Dependencies
  - None
- Exploiters
  - None

## Migration & Coexistence Considerations

- Migration
  - None
- Coexistence
  - Must be at server compatibility level 7 or greater to use this feature




In a sysplex, all the servers must be at the same server compatibility level (**serverCompatLevel** option), as determine by the first server that starts (the sysplex owner). The level must be 7 or greater to set the **pwEncryption** configuration option any of the SHA-2 or Salted SHA-2 algorithms. It must also be set to 7 for the LDAP server to accept tagged hashed SHA-2 or Salted SHA-2 hashed **userPassword** and **ibm-slapdAdminPw** attribute values on add or modify operations.

## Installation

- None

Section

# *Kerberos client updates and INADDR_ANY/in6addr_any listen option support*

## Overview (1 of 2)

- Problem Statement / Need Addressed
  - z/OS LDAP client Kerberos did not work with Active Directory Server
  - INADDR_ANY/in6addr_any would automatically bind to each interface in server
- Solution
  - Allow z/OS LDAP client to work with Active Directory server with Kerberos
  - Add INADDR_ANY/in6addr_any listen support
- Benefit / Value
  - Enables z/OS LDAP client applications to perform Kerberos binds to Active Directory Server
  - Enables Communications Server to automatically listen on all IPV4 or IPV6 interfaces on the system

## Overview (2 of 2)

- Kerberos client updates are internal only code updates – No externals affected
- INADDR_ANY and in6addr_any updates
  - Allows the z/OS Communications Server to automatically bind and listen on all configured INADDR_ANY and in6addr_any interfaces
    - Previously the LDAP server would find all active and available IP interfaces and bind and listen on those interfaces




The listen configuration option has been updated to support allowing the LDAP server to bind explicitly to the INADDR_ANY and in6addr_any interfaces on the system. This allows the Communications Server to handle binding to all interfaces available on the system instead of having the LDAP server find all active and available TCP/IP interfaces.

## Usage & Invocation (1 of 2)

- The listen option has been updated to support two new keywords: INADDR_ANY and in6addr_any
  - listen ldap://INADDR_ANY:888
    ```
    GLD1059I Listening for requests on 0.0.0.0 port 888.
    ```
  - listen ldap://in6addr_any:777
    ```
    GLD1059I Listening for requests on :: port 777.
    ```
- Previous listen support which differs from specifying no IP address or hostname on the listen option
  - listen ldap://:888
    ```
    GLD1059I Listening for requests on 2002:90c:f01:541:9:12:47:75
    port 888.
    GLD1059I Listening for requests on fe80::14:5e00:e9b7:57ef port
    888.
    GLD1059I Listening for requests on ::1 port 888.
    GLD1059I Listening for requests on 9.12.47.75 port 888.
    GLD1059I Listening for requests on 127.0.0.1 port 888.
    ```




The first listen example on this slide instructs the LDAP server to bind and listen for requests on the INADDR_ANY interface on non-secure port 888. This instructs Communications Server to automatically find all Ipv4 interfaces on the system.

The 2nd listen example on this slide instructs the LDAP server to bind and listen for requests on the in6addr_any interface on the secure port of 777. This instructs Communications Server to automatically find all Ipv4 and Ipv6 interfaces on the system.

The 3rd listen example on this slide instructs the LDAP server to find all active and available IP interfaces on the system and to bind and listen for requests explicitly on each interface. Note the differences in the server startup messages.

## Usage & Invocation (2 of 2)

– listen ldap://[::]:777

```
GLD1059I Listening for requests on 2002:90c:f01:541:9:12:47:75
port 777.
GLD1059I Listening for requests on fe80::14:5e00:e9b7:57ef port
777.
GLD1059I Listening for requests on ::1 port 777.
GLD1059I Listening for requests on 9.12.47.75 port 777.
GLD1059I Listening for requests on 127.0.0.1 port 777.
```




The listen example on this slide instructs the LDAP server to find all active and available IP interfaces on the system and to bind and listen for requests explicitly on each interface. Note the differences in the server startup messages.

## Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

## Migration & Coexistence Considerations

- Migration
  - None
- Coexistence
  - None

## Installation

- None

Section

# *Administrative group and roles*

## Overview (1 of 4)

- Problem Statement / Need Addressed
  - Administrator access should not be limited to one DN, password, instead there must be an administrative group implicitly available
  - Not all administrators are created equal
- Solution
  - Provide a mechanism to add members to the administrative group
  - Define administrator roles with different authorities such that they can be assigned to administrative group members accordingly (note roles not assigned to the group collectively, but each member)
- Benefit / Value
  - Administrator access can be controlled more effectively
  - Reduce the risk of compromising root administrator access

## Overview (2 of 4)

- Using this feature you can:
  - Define administrative group members
  - Assign different administrator roles (predefined within server) to each administrative group member
    - RACF can be used to assign roles OR
    - Entries under cn=AdminGroup,cn=Configuration in CDBM can be used to assign roles
  - User type extended operation to determine roles
- Value:
  - Root administrator credentials not needed to carry out certain administrator type tasks
  - Shared administrator credentials avoided when more than one administrator is needed
  - Users can see the administrator roles they are assigned

## Overview (3 of 4)

- New administrative roles now supported:
  - Server Configuration Group member: Administers entries under and including cn=configuration in the CDBM backend
  - Directory Data administrator: Administers all TDBM and LDBM backend entries and entries that reside under the cn=ibmpolicies suffix in the CDBM backend.
  - No administrator: Intended to quickly revoke administrative rights
  - Operational administrator: Allowed to specify the PersistentSearch control
  - Password administrator: Administers user passwords (for example allowing a help desk to reset a user's password in the LDBM and TDBM backends)
  - Replication administrator: Administers advanced replication configurations

## Overview (4 of 4)

- New administrative roles now supported (continued):
  - Root administrator: Super user authority (equivalent to adminDN in configuration file)
  - Schema administrator: Administers the server's schema

## Usage & Invocation (1 of 3)

- Must have serverCompatLevel 7 in configuration file
- CDBM must be configured in configuration file
  - ibm-slapdAdminGroupEnabled attribute in cn=configuration must be set to TRUE
  - RACF: ibm-slapdSAFSecurityDomain attribute must set in cn=configuration (DEFAULT GLDSEC)
- Bind DNs must be added to the Administrative Group
  - RACF: Bind DNs must be added to the cn=SAFAdminGroup,cn=Configuration **member** attribute value list
    - DNs when bound must resolve to a RACF ID
      - Native auth, SDBM DN, Cert MAP in RACF, Kerberos Principal MAP in RACF
  - TDS: New objectclass ibm-slapdAdminGroupMember objects must be added under cn=AdminGroup,cn=Configuration for every Bind DN

## Usage & Invocation (2 of 3)

- Administrative roles must be assigned
  - RACF: RACF ID (resolved during bind) must be given READ authority on <*securityDomain*>.ADMINROLE.<*role*> in RACF where:
    - <*securityDomain*> is set by ibm-slapdSAFSecurityDomain value from cn=configuration entry (default -> GLDSEC)
    - <*role*> one of the following: CONFIG, DIRDATA, NOADMIN, OPER, PASSWD, REPL, SCHEMA
  - TDS: ibm-slapdAdminGroupMember objects have ibm-slapdAdminRole attribute where roles are set

## Usage & Invocation (3 of 3)

- Support added for User type extended operation
  - Allows users and administrator to query their type and roles
- Examples:
  - `ldapexop -p 389 –h local host -D cn=rootadmin -w secret -op getusertype`

```
User : root_administrator
Role(s) : operational_administrator schema_administrator
replication_administrator password_administrator
directory_data_administrator server_config_administrator
```

  - `ldapexop -p 389 –h local host -D cn=master -w secret -op getusertype`

```
User    : master_server_dn
Role(s) : directory_data_administrator schema_administrator
```

## Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

## Migration & Coexistence Considerations

- Migration
  - None
- Coexistence
  - Must be at server compatibility level 7 or greater to use this feature

## Installation

- None

## Session summary

- Explain each line item in IBM TDS for z/OS
- Defined the functional content and benefit
- Identified migration, coexistence, and installation consideration

## Appendix - References

- LDAP publication references:
  - IBM Tivoli Directory Server Administration and Use for z/OS (SC23-5191)
  - IBM Tivoli Directory Server Messages and Codes for z/OS (SA23-2262)
  - IBM Tivoli Directory Server Plug-in Reference for z/OS (SA76-0148)
  - IBM Tivoli Directory Server Client Programming for z/OS (SA23-2214)

## Trademarks, disclaimer, and copyright information