z/OS LDAP: Understanding Your Server Options SHARE Session 2965

Jon Cottrell

IBM z/OS LDAP Development

jcottrel@us.ibm.com

Disclaimer

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as-is" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

Any performance data contained in this document was determined in a controlled environment and therefore, the results which may be obtained in other operating environments may vary significantly.

Users of this document should verify the applicable data for their specific environments. It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country or not yet announced by IBM. Such references or information should not be construed to mean that IBM intends to announce such IBM products, programming, or services.

Trademarks

The following are trademarks of the IBM Corporation. An asterisk following the name denotes a registered trademark.

DB2*
IBM*
OS/390
Parallel Sysplex
RACF
Tivoli
z/OS
zSeries

The names listed below are trademarks or registered trademarks and are the properties of their respective companies.

JNDI is a trademark registered by Sun Microsystems, Inc.
Kerberos is a trademark of MIT
Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
UNIX is a registered trademark of The Open Group
Windows is a trademark of Microsoft, Inc.

All statements regarding IBM's future intent are subject to change without notice, and represent goals and objectives only.

Abstract

- This presentation explains:
  - Capabilities of the IBM TDS (Tivoli Directory Server) for z/OS
  - Differences between the IBM TDS and Integrated Security Services (ISS) LDAP servers
  - Migrating data from the ISS LDAP server to IBM TDS
  - Sharing data in a sysplex environment

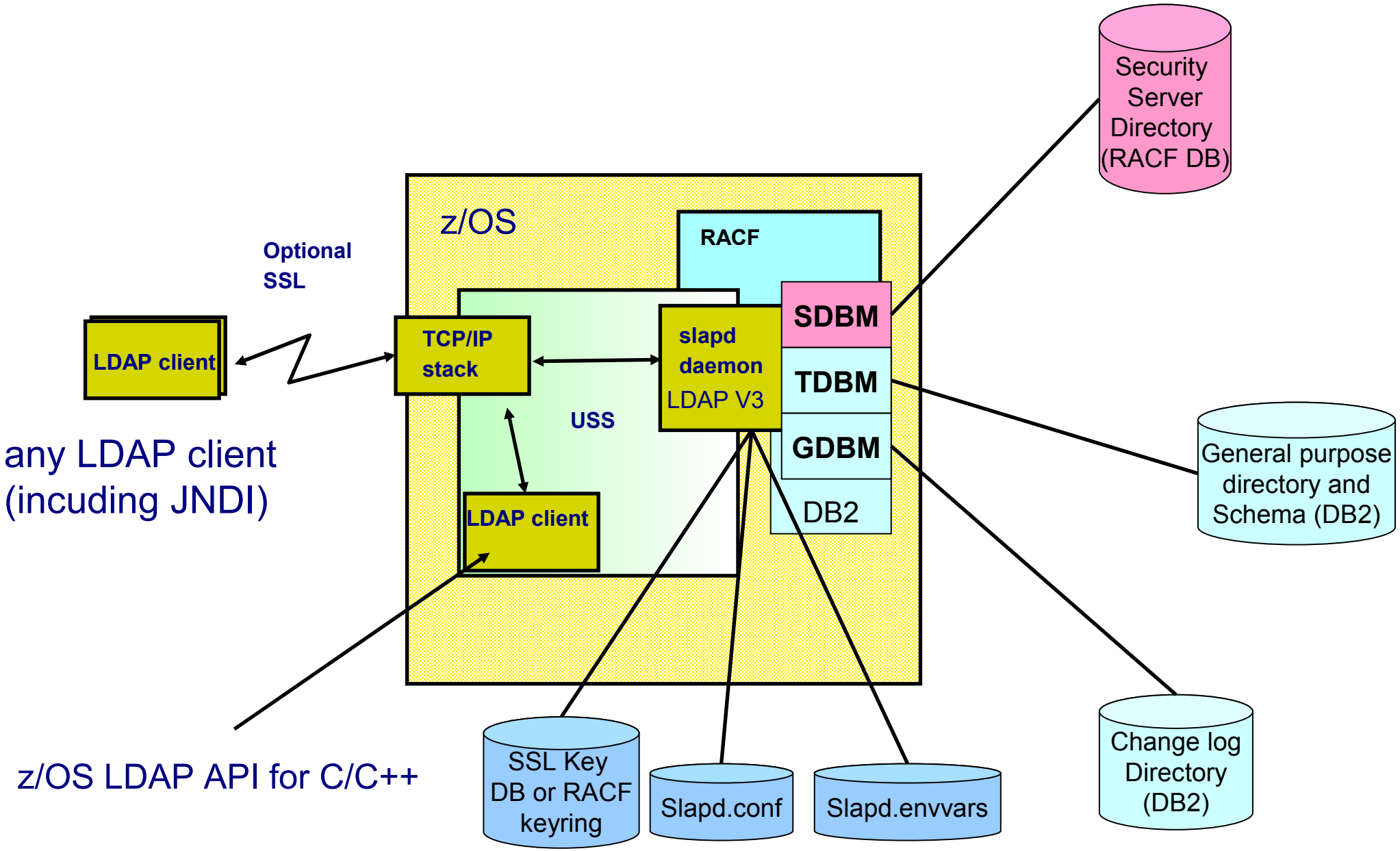
Packaging of the LDAP servers in z/OS V1R8

- Two LDAP components in z/OS V1R8:
  - **IBM Tivoli Directory Server (IBM TDS) for z/OS**
    - Enhanced (New) LDAP server
      - Available on z/OS V1R8 as of March 30th, 2007
      - Enablement APAR OA19286
    - LDAP client and utilities
  - **Integrated Security Services (ISS)**
    - z/OS V1R6 version of LDAP server
    - Coexistence PTF UA25909 to run on z/OS V1R8
- Both servers installed as part of z/OS V1R8 base
- All future enhancements to IBM TDS only

z/OS ISS LDAP Server Overview

- ISS LDAP Server has multiple backends (data stores)
  - **TDBM**: General purpose directory
    - Full LDAP V3 support, including modifiable schema
    - Data stored in DB2 database
    - Full scalability
  - **SDBM**: RACF users, groups, and user-group connections
    - Provides remote RACF administration and authentication
    - Fixed schema
    - Data stored in RACF database
    - Limited search capability
  - **GDBM**: Change log directory
    - Similar to TDBM (DB2 based) but restricted operations
    - Contains records of changes to other backends and RACF

z/OS ISS LDAP Server Overview - Continued

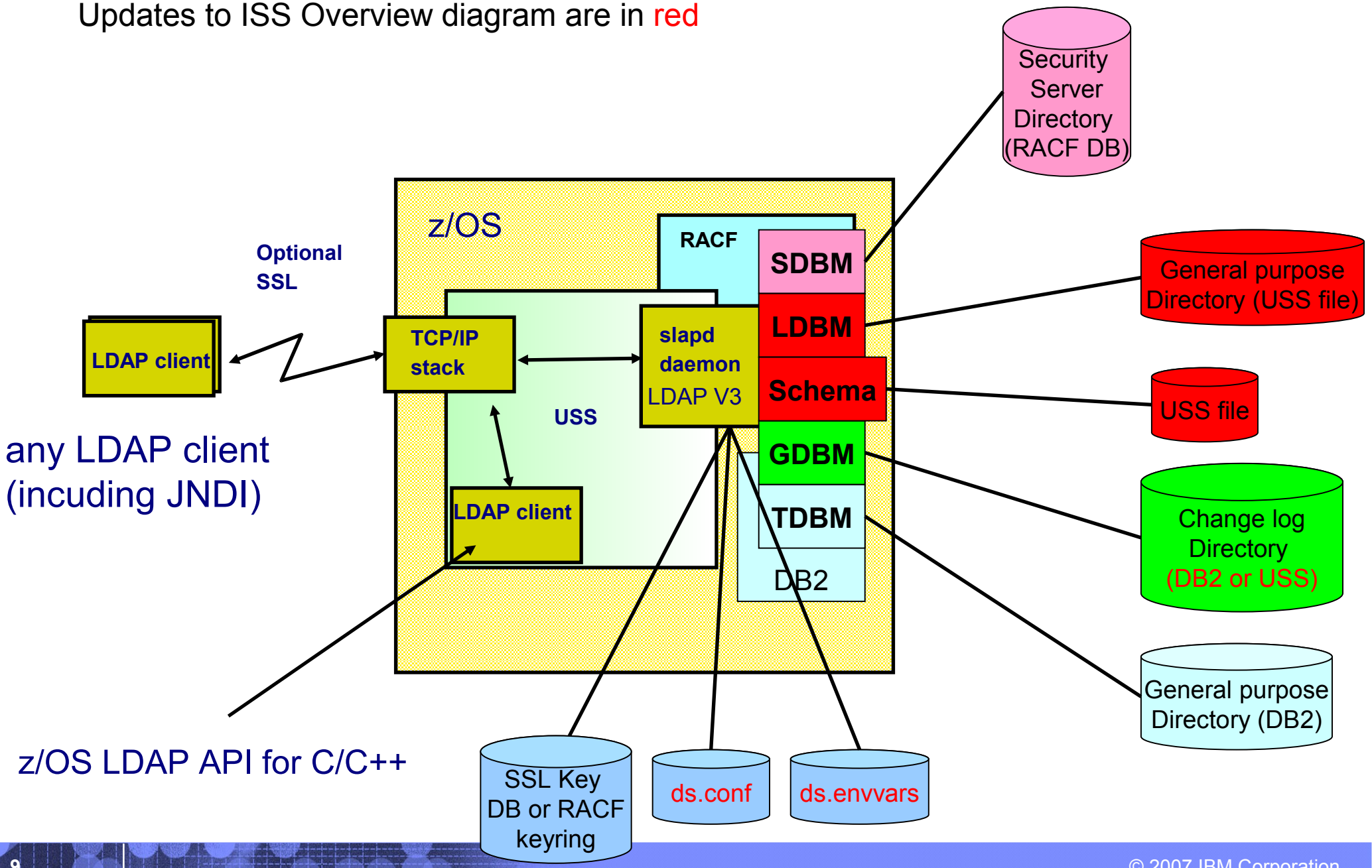


IBM TDS Overview

- Runs in 31 or 64 bit mode (if using DB2-based backends only 31 bit)
- Single server-wide schema used by all backends simplifies administration of server
- LDBM - a general purpose file-based directory
  - Uses USS file system to store directory entries
  - Does not use DB2
- GDBM (changelog backend) can be file-based or DB2-based
- SDBM (RACF-based directory) enhancements
  - Uses the R_admin profile extract functions
  - Supports new RACF command keywords

IBM TDS Overview - Continued

Updates to ISS Overview diagram are in red



Differences Between IBM TDS and ISS

- Schema differences
  - ISS LDAP server
    - Each TDBM backend has its own schema entry, schema name is cn=schema,<suffix>
    - SDBM backend has its own hardcoded schema entry
  - IBM TDS
    - Single server-wide schema used by all backends, schema name is **cn=schema**
    - Schema is stored in a USS file (**schemaPath** configuration option)
    - Non-numeric OIDs are supported in attribute types and objectclasses
    - Not allowed to change attribute types and objectclasses in use by the directory

Differences Between IBM TDS and ISS

- TDBM differences
  - ISS LDAP server
    - Schema stored within TDBM
    - LDAP replication uses the following set of DB2 tables – DIR_REGISTER, DIR_CHANGE, DIR_LONGCHANGE, and DIR_PROGRESS
    - DB_VERSION ‘3.0’ supported
  - IBM TDS
    - Schema now stored in global schema USS file
    - LDAP replication uses the following set of DB2 tables – DIR_REPLICA, DIR_REPENTRY, and DIR_LONGREPENTRY
    - DIR_MISC table updated to include a SCHEMA_TIMESTAMP column
    - DB_VERSION ‘4.0’ supported
      - Leave at DB_VERSION ‘3.0’ when sharing with ISS LDAP
  - Migration
    - ISS TDBM database can be migrated to IBM TDS

Differences Between IBM TDS and ISS - Continued

- SDBM differences
  - ISS LDAP server
    - Uses R_admin run-command interface, which has a limit of 4096 lines of output, to issue LISTUSER/LISTGRP commands
    - Limit can cause incomplete LISTUSER/LISTGRP output resulting incomplete group gathering when performing an SDBM bind
    - R_admin run-command interface returns ‘no value’ if RACF keyword is set to ‘no value’
  - IBM TDS
    - Uses R_admin profile extract interface for obtaining RACF users, groups, and user-group connections
    - No limit on the output from the R_admin profile extract interface
    - R_admin profile extract does NOT return ‘no value’ if RACF keyword if set to ‘no value’
- RACF still limits search output when using the RACF SEARCH command in both servers

Differences Between IBM TDS and ISS - Continued

- LDBM (file-based backend) – TDS only
  - General purpose backend (like TDBM) – stores any data
    - Supports TDBM functionality – alias, native authentication, replication
    - Entries stored in USS files (DB2 not used)
      - Separate database file for each LDBM suffix
      - Single checkpoint file contains updates to entries in database files
      - Replication files store replication info
    - Can change directory where LDBM files are stored with **databaseDirectory** configuration option
    - Multiple LDBM backends can be configured
    - Runs in 31 and 64 bit mode
    - Can be shared in a sysplex

Differences Between IBM TDS and ISS - Continued

- LDBM (file-based backend) – Continued
  - Entries stored in memory while server is running
    - Advantage: fast access
    - Trade off: large directory requires a lot of memory and server start-up and shutdown is slow (reads in all those entries)
    - Intended for small to medium size directory
      - About 250K entries in 31 bit mode
      - About 500K entries in 64 bit mode

Differences Between IBM TDS and ISS - Continued

- GDBM (Change log directory) differences
  - ISS LDAP server
    - DB2-based only
  - TDS
    - Can be file or DB2 based
      - File-based can run in 31 or 64 bit mode
      - DB2-based only runs in 31 bit mode

Differences Between IBM TDS and ISS - Continued

- Reliability, Availability, and Serviceability (RAS)
  - ISS LDAP server
    - Configuration file errors are tolerated although server would start when one or more backends would fail configuration
  - IBM TDS
    - Configuration file errors can now fail startup if one more backends fail to configure
    - SMF 83 audit records are created for server events
      - RACF SMF unload supports LDAP audit records
      - DB2 samples and XML schema provided in **GLD.SGLDSAMP**
      - Controlled by a configuration option or operator command

Differences Between IBM TDS and ISS - Continued

- Reliability, Availability, and Serviceability (RAS) - Continued
  - ISS LDAP server
    - In-memory trace used for the logging of some internal server errors
  - IBM TDS
    - ARM support available
      - LDAP server registers with ARM during server startup
    - LDAP server creates CTRACE records even if DEBUG is not on or DEBUG ERROR is not set
      - CTRACE records in dump can be viewed with IPCS
      - Debug can be directed to CTRACE only

Differences Between IBM TDS and ISS - Continued

- Reliability, Availability, and Serviceability (RAS) - Continued
  - ISS LDAP server
    - DB2 monitoring available
    - File system monitoring and TCP/IP monitoring not available
  - IBM TDS
    - DB2 monitoring available – similar to ISS LDAP server
    - File system monitoring – used with LDBM and GDBM (file-based) backends
    - TCP/IP monitoring available
- Miscellaneous Support differences
  - ISS LDAP server
    - Sample started task proc (GLDSRV) runs GLDSLAPD
  - IBM TDS
    - Sample started task proc (DSSRV) runs GLDSRV31 or GLDSRV64
    - ldapcompare utility shipped

Differences Between IBM TDS and ISS - Continued

- Miscellaneous Support differences - Continued
  - ISS LDAP server
    - crypt, SHA, MD5, and DES encryption supported for **userPassword** attribute values with the **pwEncryption** configuration option
    - **ldapcnf** – LDAP configuration utility
    - **tdbm2ldif** – TDBM and schema unload utility
    - **ldif2tdbm** – TDBM bulkload utility
  - IBM TDS
    - crypt, SHA, MD5, DES, and AES encryption supported for **userPassword** attribute values with the **pwEncryption** configuration option
    - DES and AES encryption of **secretKey** values with the new **secretEncryption** configuration option
    - NULL-based search supported – allows searching LDBM and TDBM backends (SDBM does not participate)
    - **dsconfig** – LDAP configuration utility
    - **ds2ldif** – TDBM, LDBM, and schema unload utility
    - **ldif2ds** – TDBM bulkload utility, schema must be modified beforehand

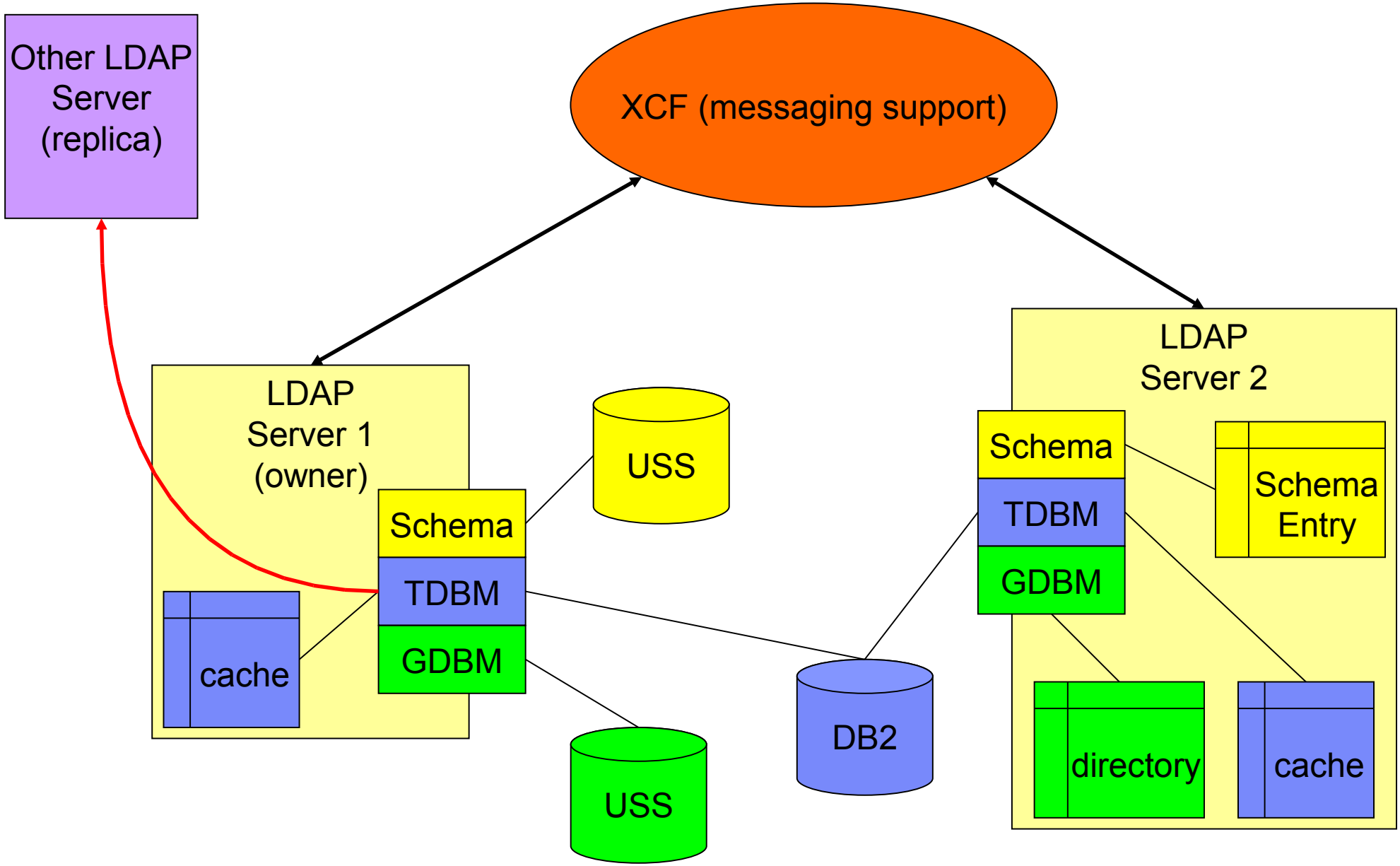
Differences Between IBM TDS and ISS - Continued

- Replication Support differences
  - ISS LDAP server
    - All replication for all backends (TDBM only) are controlled by the server as a whole
    - Failing operations can be set aside (APAR OA14456)
  - IBM TDS
    - Supported in a sysplex – uses XCF services to coordinate replication of updates
    - Controlled by each backend (TDBM and LDBM)
      - Individual backend can be a master server, peer server, read-only replica, or not participate in replication
      - Determined by **masterServerDN** or **peerServerDN** configuration options for backend
      - Replica entry in backend directory
    - Failing operations can be set aside

Differences Between IBM TDS and ISS - Continued

- Sysplex Support differences
  - ISS LDAP server
    - Search cache support is disabled in multiserver mode
    - Only used DB2 datasharing – XCF messaging not used
    - LDAP replication only allowed as a read-only consumer/replica
  - IBM TDS
    - Equivalent cache support in single and multiserver mode
    - Event notification via XCF messaging
    - LDAP replication allowed as a read/write master or peer-to-peer in a sysplex

IBM TDS Sysplex Support



IBM TDS Sysplex Support - Continued

- First server in sysplex is the owner - Owns the resources
  - USS files (Schema, LDBM, and file-based GDBM)
    - Performs update and sends change to sysplex replicas
    - USS files must be in either zFS or shared HFS and available to all servers in the sysplex
  - Owns LDAP replication out of the sysplex
- Other servers are sysplex replicas
  - LDBM request directory from owner at startup
  - Applies changes from owner
  - Passes modification requests to owner

IBM TDS Sysplex Support - Continued

- TDBM and GDBM (DB2-based) backends are shared between an sysplex replicas
  - Uses DB2 data sharing
  - Notifies other LDAP servers of changes
  - Invalidates internal caches when notified of changes

- When owning server terminates
  - Oldest sysplex replica becomes owner
  - Owns resources
  - Replicates

Migration and Coexistence Considerations

- **Migration**:
  - Can migrate TDBM data in ISS LDAP server to a TDBM or LDBM backend in IBM TDS
  - New messages and reason codes in IBM TDS
  - New operator command syntax in IBM TDS
    - f name,type [options]
- **Coexistence**:
  - Share TDBM data between ISS LDAP server and IBM TDS
    - Some restrictions on both the ISS and IBM TDS
    - Requires coexistence PTF UA25909 on ISS LDAP server

Migration Considerations

- Configuration file changed
  - Example: DLL names and sysplex configuration options
  - Can use **dsconfig** to create new files and server proc
- TDBM schema migration
  - Done automatically by IBM TDS during initialization
  - Problem if multiple TDBM backends in ISS server with conflicting schema definitions

Migration Considerations - Continued

- Replication queue cannot be migrated
  - Make sure ISS LDAP server replication is completed before migration
- Replication in IBM TDS is controlled by each backend
  - May need to add additional replica entries after migration if migrating multiple TDBM backends

Migration Considerations - Continued

- Migrating ISS LDAP TDBM to IBM TDS TDBM
  - Add TDBM options to IBM TDS server configuration file
  - Run TDBMMGRT SPUFI script to update ISS TDBM tables to IBM TDS TDBM level
    - Creates new replication tables
    - Optionally updates DB_VERSION to ‘4.0’
      - Do not do if sharing TDBM with ISS LDAP server
  - Start IBM TDS
    - IBM TDS LDAP server merges TDBM backend schema into server global schema
  - Uses TDBM data entries without any migration

Migration Considerations - Continued

- Migrating ISS LDAP TDBM to IBM TDS LDBM
  - Add LDBM options to IBM TDS server configuration file
  - Unload schema from ISS LDAP TDBM backend and load into IBM TDS global schema
    - Use **tdbm2ldif** to unload, **ldapmodify** to load
    - Need to edit unload output to remove suffix in DN
      - Note – do not use editor that removes blanks
  - Unload entries from ISS LDAP TDBM backend and load into IBM TDS LDBM backend
    - Use **tdbm2ldif** to unload, **ldapadd** to load

Migration Considerations - Continued

- Migrating ISS LDAP GDBM to IBM TDS DB2-based GDBM
  - Add DB2-based GDBM options to IBM TDS configuration file
  - No LDAP unload – load required – can use data as is

- Converting ISS LDAP GDBM to IBM TDS file-based GDBM
  - Ensure all entries in ISS LDAP server changelog have been processed before conversion
  - Add file-based GDBM options to IBM TDS configuration file

Sharing Data between IBM TDS and ISS LDAP Server

- ISS LDAP V1R6 or later and IBM TDS can share TDBM and GDBM (DB2-based) databases
  - Configure sysplex and multiserver on each server
  - DB_VERSION must be ‘3.0’ (not ‘4.0’)
  - Function limited to ISS server
    - Enhanced sysplex support and replication disabled
    - Schema updates that contain Non-numeric OID rejected
  - ISS LDAP requires Coexistence PTF UA25909

- ISS LDAP server restrictions
  - All schema updates must be to IBM TDS global schema
    - IBM TDS LDAP server pushes schema updates to TDBM backend schema

Session Summary

- Overview of the ISS LDAP server and the IBM TDS
- Explained packaging of the LDAP servers within z/OS
- Differences between the ISS LDAP server and the IBM TDS
- Identified migration and coexistence considerations

Publications References

- 'New' IBM TDS
  - SC23-5191 IBM Tivoli Directory Server Server Administration and Use for z/OS
    - Available on z/OS V1R8 as of March 31, 2007
- 'New' IBM TDS LDAP client
  - SA23-2214 IBM Tivoli Directory Server Client Programming for z/OS
    - Available at z/OS V1R8 GA
- 'Old' Integrated Security Services LDAP server
  - SC24-5923 z/OS Integrated Security Services LDAP Server Administration and Use
    - Available at z/OS V1R8 GA

z/OS LDAP:
Understanding Your Server Options
SHARE Session 2965

Jon Cottrell
IBM z/OS LDAP Development
jcottrel@us.ibm.com

Disclaimer

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as-is" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

Any performance data contained in this document was determined in a controlled environment and therefore, the results which may be obtained in other operating environments may vary significantly.

Users of this document should verify the applicable data for their specific environments. It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country or not yet announced by IBM. Such references or information should not be construed to mean that IBM intends to announce such IBM products, programming, or services.

Trademarks

The following are trademarks of the IBM Corporation. An asterisk following the name denotes a registered trademark.

DB2*
IBM*
OS/390
Parallel Sysplex
RACF
Tivoli
z/OS
zSeries

The names listed below are trademarks or registered trademarks and are the properties of their respective companies.

JNDI is a trademark registered by Sun Microsystems, Inc.
Kerberos is a trademark of MIT
Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
UNIX is a registered trademark of The Open Group
Windows is a trademark of Microsoft, Inc.

All statements regarding IBM's future intent are subject to change without notice, and represent goals and objectives only.

Abstract

- This presentation explains:
  - Capabilities of the IBM TDS (Tivoli Directory Server) for z/OS
  - Differences between the IBM TDS and Integrated Security Services (ISS) LDAP servers
  - Migrating data from the ISS LDAP server to IBM TDS
  - Sharing data in a sysplex environment

From this point onward in this presentation, the z/OS Integrated Security Services LDAP server will be referred to as ISS. While the IBM Tivoli Directory Server for z/OS will be referred to as IBM TDS.

Packaging of the LDAP servers in z/OS V1R8

- Two LDAP components in z/OS V1R8:
  - **IBM Tivoli Directory Server (IBM TDS) for z/OS**
    - Enhanced (New) LDAP server
      - Available on z/OS V1R8 as of March 30th, 2007
      - Enablement APAR OA19286
    - LDAP client and utilities
  - **Integrated Security Services (ISS)**
    - z/OS V1R6 version of LDAP server
    - Coexistence PTF UA25909 to run on z/OS V1R8
- Both servers installed as part of z/OS V1R8 base
- All future enhancements to IBM TDS only

The IBM Tivoli Directory Server is the new LDAP component of z/OS V1R8. As of March 30th, 2007 the IBM TDS has been generally available. In order to activate IBM TDS, it is necessary to order the following PTFs associated with the enablement APAR OA19286: UA32981, UA32982, and UA32983. IBM TDS now contains both the LDAP server and client.

The 'old' LDAP server is still shipped in the Integrated Security Services component of z/OS V1R8. In z/OS V1R8, the LDAP client is no longer shipped with ISS.

The coexistence PTF UA25909 allows the ISS LDAP server to run on z/OS V1R8.

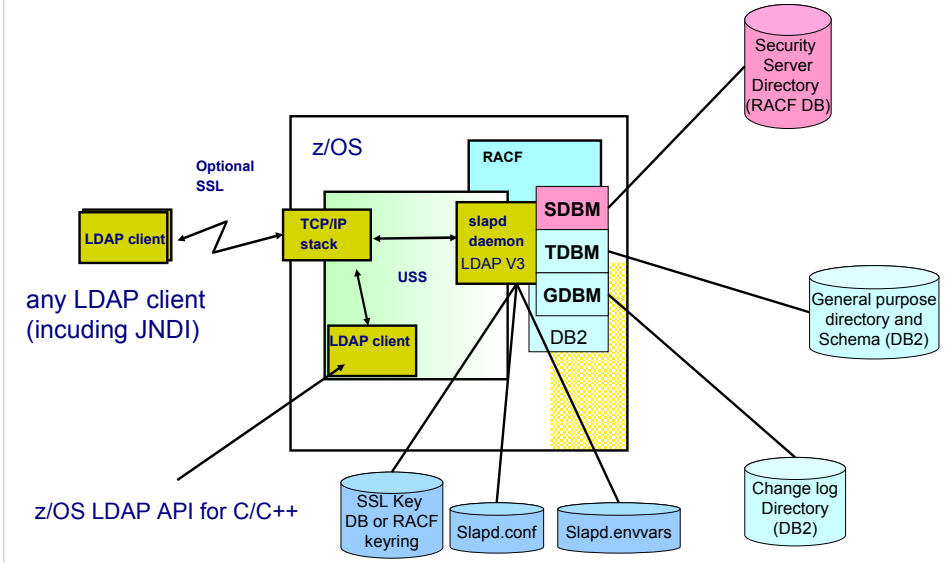
z/OS ISS LDAP Server Overview

- ISS LDAP Server has multiple backends (data stores)
  - **TDBM**: General purpose directory
    - Full LDAP V3 support, including modifiable schema
    - Data stored in DB2 database
    - Full scalability
  - **SDBM**: RACF users, groups, and user-group connections
    - Provides remote RACF administration and authentication
    - Fixed schema
    - Data stored in RACF database
    - Limited search capability
  - **GDBM**: Change log directory
    - Similar to TDBM (DB2 based) but restricted operations
    - Contains records of changes to other backends and RACF

Restricted sysplex get only DB2 database sharing and limited caching support within the server.

The limited search capability for the SDBM backend indicates that there are a limited number of search filters that are supported in the SDBM backend (i.e. objectclass=*).

z/OS ISS LDAP Server Overview - Continued



Any LDAP client, including JNDI, which supports the LDAP V2 or V3 protocol can communicate with the ISS LDAP server.

The z/OS LDAP client will talk to any LDAP server that supports LDAP V2 or V3 protocol.

IBM TDS Overview

- Runs in 31 or 64 bit mode (if using DB2-based backends only 31 bit)
- Single server-wide schema used by all backends simplifies administration of server
- LDBM - a general purpose file-based directory
  - Uses USS file system to store directory entries
  - Does not use DB2
- GDBM (changelog backend) can be file-based or DB2-based
- SDBM (RACF-based directory) enhancements
  - Uses the R_admin profile extract functions
  - Supports new RACF command keywords

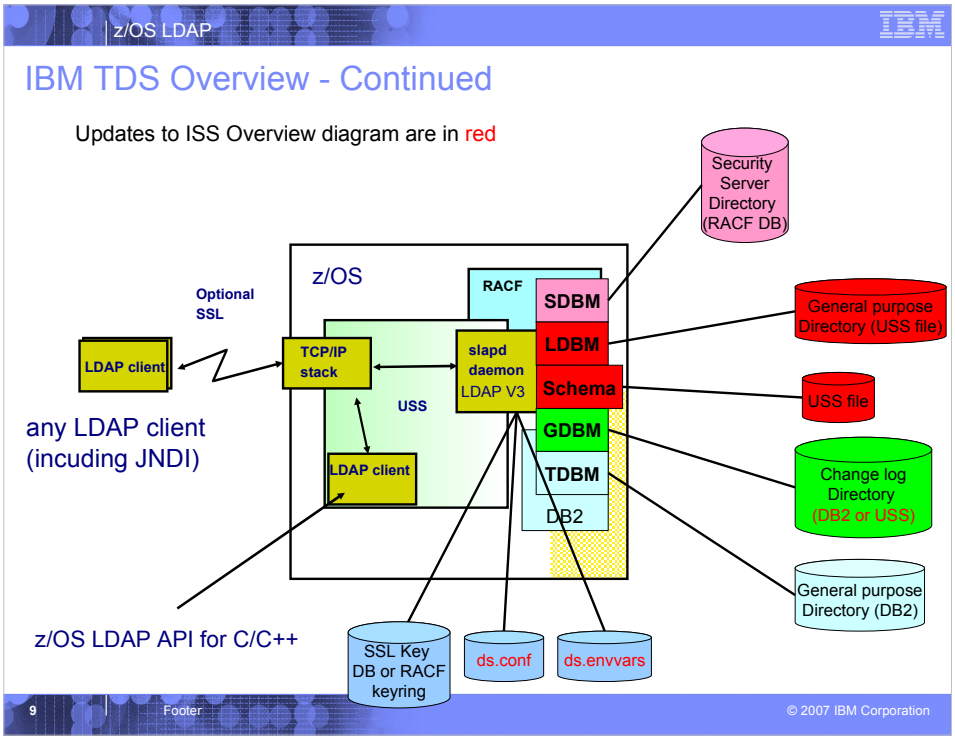
A backend is a part of the LDAP server that administers a set of directory entries. The backends differ in what type of directory entries they can store and where they store the entries. The IBM TDS supports all the backends which were supported by the ISS LDAP server..

In the ISS LDAP server, each TDBM and SDBM backend had its own individual schema, stored within the backend. The schema indicated what type of data the backend could store. As the number of backends expanded, this became increasing confusing and difficult to manage. The IBM TDS has a single schema entry for the entire server which is shared across all backends. The schema is stored in a USS file.

LDBM is the new backend that stores its entries in files. It is comparable to TDBM in the types of data it can store and the functions that it provides, but does not use DB2.

In the ISS LDAP server, GDBM is required DB2 to store change log entries. In IBM TDS, the GDBM backend can now be configured to store its entries within a USS file or DB2.

The SDBM backend has been enhanced to use the R_admin profile extract interface to avoid, in some cases, the limitations on the amount of data RACF returns.



On an earlier slide in this presentation, the ISS LDAP server was depicted in a similar diagram. This diagram shows IBM TDS. The following are the major differences between the two servers and they are also highlighted in red in the above diagram:

2. The LDBM backend has been introduced in IBM TDS. This is a general purpose directory where the data is stored out in a USS file system.
3. The schema data is now stored within a separate backend out in a USS file system. In ISS, each TDBM backend had its own schema which was stored in DB2. The SDBM backend's schema was hardcoded within SDBM. Since there is now a backend, the schema is now global across all backends (LDBM, TDBM, and SDBM).
4. The GDBM backend can be stored in either DB2 (TDBM-based) or a USS (LDBM-based) file system.
5. The default server configuration file is now ds.conf and the default environment file is now called ds.envvars.

Any LDAP client, including JNDI, which supports the LDAP V2 or V3 protocol can communicate with the IBM TDS.

The z/OS LDAP client will talk to any LDAP server that supports LDAP V2 or V3 protocol.

Differences Between IBM TDS and ISS

- Schema differences
  - ISS LDAP server
    - Each TDBM backend has its own schema entry, schema name is cn=schema,<suffix>
    - SDBM backend has its own hardcoded schema entry
  - IBM TDS
    - Single server-wide schema used by all backends, schema name is **cn=schema**
    - Schema is stored in a USS file (**schemaPath** configuration option)
    - Non-numeric OIDs are supported in attribute types and objectclasses
    - Not allowed to change attribute types and objectclasses in use by the directory

The schema is now named cn=schema. The schema entry is in its own backend and is stored in a file in the Unix System Services file system, with a name of **schema.db**. By default the file is stored in the /var/ldap/schema directory, but the directory can be changed using the **schemaPath** configuration option. The file name cannot be changed. The LDAP server must have read/write access to the schema file and directory.

When the IBM TDS is first started, an initial schema is automatically created by the LDAP server. This schema contains all of the attributes and object classes used by GDBM and SDBM plus the attributes and object classes used by some TDBM and LDBM functions (such as native authentication, referrals, aliases). The initial schema probably will not have the attributes and object classes needed for a customer's usage of TDBM or LDBM. In that case, the customer will need to add additional attributes and object classes to the schema. The attributes and object classes in the initial schema cannot be changed or deleted.

The IBM TDS schema now supports non-numeric OIDs, like some other LDAP servers. Such an OID looks like:

attributetypes: (myNewAttr-oid NAME ('myNewAttr') DESC 'New attribute for Project X' SYNTAX 1.3.6.1.4.1.1466.115.12.1.15 USAGE userApplications)

Note that non-numeric OIDs are not supported by the old ISS LDAP server. Do not use them if you plan to share TDBM between the old and new LDAP servers. See the Migration section of this presentation for more information about sharing TDBM data.

The IBM TDS does not allow "harmful" schema modifications to occur. An example of a "harmful" schema operation is one in which a 'MUST' (required) is removed from an objectclass definition that is still in use by entries within the directory. In the ISS LDAP server this is allowed to occur.

Differences Between IBM TDS and ISS

- TDBM differences
  - ISS LDAP server
    - Schema stored within TDBM
    - LDAP replication uses the following set of DB2 tables – DIR_REGISTER, DIR_CHANGE, DIR_LONGCHANGE, and DIR_PROGRESS
    - DB_VERSION '3.0' supported
  - IBM TDS
    - Schema now stored in global schema USS file
    - LDAP replication uses the following set of DB2 tables – DIR_REPLICA, DIR_REPENTRY, and DIR_LONGREPENTRY
    - DIR_MISC table updated to include a SCHEMA_TIMESTAMP column
    - DB_VERSION '4.0' supported
      - Leave at DB_VERSION '3.0' when sharing with ISS LDAP
  - Migration
    - ISS TDBM database can be migrated to IBM TDS

The IBM TDS TDBM backend has the same functionality as TDBM in the ISS LDAP server. There are some changes:

2. The TDBM replication support has been re-structured to use 3 different tables. DIR_REPLICA stores information on replica servers. DIR_REPENTRY contains the data to be replicated. DIR_LONGREPENTRY stores the data to be replicated if it is too large to fit into DIR_REPENTRY. These tables replace the old DIR_REGISTER, DIR_CHANGE, DIR_LONGCHANGE, and DIR_PROGRESS tables. The old tables are ignored if found in the IBM TDS TDBM database.
3. The DIR_MISC table now contains a SCHEMA_TIMESTAMP column. This is used when an IBM TDS is sharing a TDBM database with an ISS LDAP server, to keep the TDBM schema in ISS LDAP and the global schema in IBM TDS in sync.
4. The DB_VERSION column in the DIR_MISC table for a for a new TDBM database is set to '4.0' to indicate that it can support all of the new sysplex and replication features of the IBM TDS. This level of TDBM cannot be shared with an ISS LDAP server.

   The DB_VERSION value of a TDBM database created by an ISS LDAP server is '3.0'. This TDBM database can be shared with the IBM TDS, but it will not be able to use the enhanced sysplex and replication features. The DB_VERSION of the TDBM database can be changed to '4.0' as part of migration to the IBM TDS.
5. The schema is no longer stored in the TDBM backend. During migration of a TDBM backend from an ISS LDAP server to an IBM TDS, the TDBM schema is merged into the IBM TDS global schema. After that, the TDBM schema is no longer used. See the Migration part of this presentation for more information on TDBM migration.

Differences Between IBM TDS and ISS - Continued

- SDBM differences
  - ISS LDAP server
    - Uses R_admin run-command interface, which has a limit of 4096 lines of output, to issue LISTUSER/LISTGRP commands
    - Limit can cause incomplete LISTUSER/LISTGRP output resulting incomplete group gathering when performing an SDBM bind
    - R_admin run-command interface returns ‘no value’ if RACF keyword is set to ‘no value’
  - IBM TDS
    - Uses R_admin profile extract interface for obtaining RACF users, groups, and user-group connections
    - No limit on the output from the R_admin profile extract interface
    - R_admin profile extract does NOT return ‘no value’ if RACF keyword if set to ‘no value’
- RACF still limits search output when using the RACF SEARCH command in both servers

The SDBM backend is re-written in the IBM TDS to take advantage of the RACF R_admin profile extract interface. SDBM in the ISS LDAP server uses the R_admin run-command interface to issue LISTUSER/LISTGRP commands to obtain user, group, and connection profiles. The R_admin run-command interface has a limit of 4096 lines of output, so the output from a LISTUSER for a user who belongs to many groups or a LISTGRP for a group that contains many members is incomplete. In particular, when gathering the groups for such a user, the user’s list of access groups is incomplete, potentially resulting in incorrect access control decisions for TDBM and GDBM entries. The R_admin profile extract interface used by SDBM in the IBM TDS returns the complete user, group, or connection profile. In particular, an LDAP client binding to SDBM will have a complete list of groups with which to determine access to TDBM, LDBM, and GDBM entries.

There is another difference between use of R_admin profile extract and R_admin run-command for LISTUSER and LISTGRP. LISTUSER and LISTGRP output includes special ‘no value’ keyword values to indicate that there is no value for the keyword in the user, group, or connection profile. R_admin profile extract simply does not return the keyword when there is no value. Thus a RACF user or group entry returned by the ISS LDAP server may contain many attributes that are not in the entry returned by the new IBM TDS if the corresponding keywords in the RACF profile do not have a value.

Note that there are still some SDBM searches that use the R_admin run-command interface to issue the RACF SEARCH command. These searches continue to be affected by the RACF 4096 limit on output. An example of such as search is the filter “racfid=U*”, which searches for all RACF users whose user ID begins with U.

Differences Between IBM TDS and ISS - Continued

- LDBM (file-based backend) – TDS only
  - General purpose backend (like TDBM) – stores any data
    - Supports TDBM functionality – alias, native authentication, replication
    - Entries stored in USS files (DB2 not used)
      - Separate database file for each LDBM suffix
      - Single checkpoint file contains updates to entries in database files
      - Replication files store replication info
    - Can change directory where LDBM files are stored with **databaseDirectory** configuration option
    - Multiple LDBM backends can be configured
    - Runs in 31 and 64 bit mode
    - Can be shared in a sysplex

LDBM is a new general purpose backend provided by the IBM TDS. It stores its entries in files within the Unix System Services file system. There are two types of files used by LDBM, database files and the checkpoint file.

For each suffix in the LDBM backend, there is a database file to contain the entries under that suffix (including the suffix itself). The file name of the database files is LDBM-n.db, where n is a number assigned by the LDAP server for that suffix (thus LDBM-1.db, LDBM-2.db, etc.).

For the entire LDBM backend, there is one checkpoint file (LDBM.ckpt). Every time an LDBM entry is added, changed, renamed, or deleted, information about the operation is put in the checkpoint file. These changes are applied to the appropriate database files at a later time.

If LDBM is configured for replication, it will also use 2 additional files. The replication file, named LDBM.repl, contains information about updates to the server that need to be replicated. The progress file, named LDBM.prog, contains information about which replication operations have been processed at which replicas.

The default directory for the LDBM files is /var/ldap/ldbm. The directory can be changed using the databaseDirectory option in the LDBM backend section of the LDAP server configuration file. The file names cannot be changed. The LDAP server must have read-write access to the directory and the files. When multiple LDBM backends are configured, each backend must have its own databaseDirectory configured.

Differences Between IBM TDS and ISS - Continued

- LDBM (file-based backend) – Continued
  - Entries stored in memory while server is running
    - Advantage: fast access
    - Trade off: large directory requires a lot of memory and server start-up and shutdown is slow (reads in all those entries)
    - Intended for small to medium size directory
      - About 250K entries in 31 bit mode
      - About 500K entries in 64 bit mode

When the IBM TDS is started, all of the LDBM entries are read into memory, providing very fast access to the entries. However, a large LDBM directory will use a lot of memory and can require a long time to initialize when the server is started.

Paging hurts performance. So, how much real storage are you willing to use?

Differences Between IBM TDS and ISS - Continued

- GDBM (Change log directory) differences
  - ISS LDAP server
    - DB2-based only
  - TDS
    - Can be file or DB2 based
      - File-based can run in 31 or 64 bit mode
      - DB2-based only runs in 31 bit mode

The IBM TDS allows the changelog backend, GDBM, to be configured to store the change log entries in either DB2 or files. The use of the dbuserid option in the GDBM section of the configuration file indicates whether the GDBM backend is DB2-based; otherwise, the GDBM backend is file-based.

When file-based, GDBM files are located in /var/ldap/gdbm by default. There is a single database file, named LDBM-1.db. The checkpoint file name is LDBM.ckpt. The path can be changed using the databaseDirectory option in the GDBM backend of the configuration file. The file names cannot be changed.

An LDBM backend can be configured to not log changes to its entries using the changeLoggingParticipant option in the LDBM section of the configuration file. By default, changes are logged. Similarly, changes to the server schema are logged unless 'changeLoggingParticipant no' is specified in the GDBM section of the configuration file. Note that changes to the GDBM suffix entry or change log entries are never logged.

Differences Between IBM TDS and ISS - Continued

- Reliability, Availability, and Serviceability (RAS)
  - ISS LDAP server
    - Configuration file errors are tolerated although server would start when one or more backends would fail configuration
  - IBM TDS
    - Configuration file errors can now fail startup if one more backends fail to configure
    - SMF 83 audit records are created for server events
      - RACF SMF unload supports LDAP audit records
      - DB2 samples and XML schema provided in **GLD.SGLDSAMP**
      - Controlled by a configuration option or operator command

The **srvStartUpError** configuration option in the IBM TDS controls whether or not the LDAP server stops when a backend fails to start during server startup. Specify TERMINATE if you do not want the LDAP server to run without all of its configured backends. Specify IGNORE if you want the LDAP server to process requests for whatever backends do initialize successfully. Note that the LDAP server always stops if it detects an error in the configuration file or if the schema can not be initialized. Once the server has started, the **srvStartUpError** option no longer has any effect if a backend fails.

The IBM TDS invokes R_auditx() to create SMF audit records for operations. SMF unload has been updated to include formatting for LDAP SMF type 83 audit records. Samples for DB2 formatting of SMF data and schema for XML formatting are provided in GLD.SGLDSAMP.

LDAP server audit usage can be controlled using the **audit** configuration option in the global section of the LDAP server configuration file or the AUDIT modify operator command.

Differences Between IBM TDS and ISS - Continued

- Reliability, Availability, and Serviceability (RAS) - Continued
  - ISS LDAP server
    - In-memory trace used for the logging of some internal server errors
  - IBM TDS
    - ARM support available
      - LDAP server registers with ARM during server startup
    - LDAP server creates CTRACE records even if DEBUG is not on or DEBUG ERROR is not set
      - CTRACE records in dump can be viewed with IPCS
      - Debug can be directed to CTRACE only

The IBM TDS registers itself with the z/OS Automatic Restart Management (ARM) if the **armName** option is specified in the global section of the configuration file. ARM can be used to react to a failure of the LDAP server.

Many LDAP server errors result in issuing an LDAP debug trace record if DEBUG is active and the ERROR debug level is set. These records can be very helpful in resolving server or application problems. Unfortunately, activating the LDAP trace facility incurs an LDAP server performance penalty, even if no errors occur and no trace records are created. The IBM TDS uses CTRACE to capture error information with much less performance impact than the trace facility.

The CTRACE facility writes ERROR debug trace records to the in-memory server CTRACE buffers whenever an ERROR trace record would be created if DEBUG ERROR were set, even if DEBUG is not active or is active but the ERROR level is not set. Thus the LDAP server can be run without the DEBUG trace facility active and still capture error information. If the DEBUG trace facility is active, the trace records also get written to the normal trace output target, unless the LDAP_DEBUG_FILENAME environment variable is set to CTRACE_ONLY. In that case, all DEBUG output is only sent to the CTRACE buffers.

The CTRACE buffers can be viewed using IPCS and the GLDSCTFT formatting routine by dumping the LDAP server address space. The CTRACE facility uses wrap-around buffers: the earliest CTRACE records are overwritten when the buffers become full. The LDAP_CTRACE_BUFFSIZE environment variable can be used to set the total size of the buffers. The minimum (and default) size is 1024000 bytes.

Differences Between IBM TDS and ISS - Continued

- Reliability, Availability, and Serviceability (RAS) - Continued
  - ISS LDAP server
    - DB2 monitoring available
    - File system monitoring and TCP/IP monitoring not available
  - IBM TDS
    - DB2 monitoring available – similar to ISS LDAP server
    - File system monitoring – used with LDBM and GDBM (file-based) backends
    - TCP/IP monitoring available
- Miscellaneous Support differences
  - ISS LDAP server
    - Sample started task proc (GLDSRV) runs GLDSLAPD
  - IBM TDS
    - Sample started task proc (DSSRV) runs GLDSRV31 or GLDSRV64
    - ldapcompare utility shipped

The **db2Terminate** configuration option specifies how the LDAP server will react to a termination of DB2. If the **db2Terminate** configuration option is set to TERMINATE, the LDAP server will shutdown if DB2 terminates. If the **db2Terminate** configuration option is set to RECOVER or RESTORE, the LDAP server will disconnect from DB2 and acesss to the DB2 backends are rejected with an error. Once DB2 is running again, the LDAP server will reconnect to DB2 and access will be restored.

The **fileTerminate** configuration option determines what the IBM TDS does when an file error is encountered in the LDBM or GDBM (file-based) backends. If the **fileTerminate** configuration option is set to TERMINATE, the LDAP server will shutdown when an file error is encountered (i.e. out of space in the file system) in the LDBM or GDBM (file-based) backend. If the **fileTerminate** configuration option is set to RECOVER, the LDAP server will put the LDBM or GDBM backend into readOnly mode until the file system error can be recovered. Once the file system error is recovered from, an LDAP server modify command can put it back into read/write mode. The default is RECOVER.

The IBM TDS network interface support contains a monitor that polls the TCP/IP stacks that the LDAP server is listening on to determine when they go down and come back up. This polling defaults to once every 5 minutes, but the interval can be changed with the LDAP_NETWORK_POLL environment variable. If an interface fails but the LDAP server still has at least one active interface, the server continues processing and re-establishes a failed interface when it detects that it has become active. If all the interfaces fail, the response of the LDAP server depends on the **tcpTerminate** option in the global section of the LDAP server configuration file: the server can stop (TERMINATE) or the server can stay up and re-establish network interfaces when they become active (RECOVER). The default is RECOVER.

The IBM TDS DSSRV sample proc supports running the server in 31 or 64 bit mode. The GLDSRV31 executable is the 31 bit version of the server, while GLDSRV64 is the 64 bit version of the server. The ldapcompare utility is now shipped on z/OS V1R8 as part of the IBM TDS.

Differences Between IBM TDS and ISS - Continued

- Miscellaneous Support differences - Continued
  - ISS LDAP server
    - crypt, SHA, MD5, and DES encryption supported for **userPassword** attribute values with the **pwEncryption** configuration option
    - **ldapcnf** – LDAP configuration utility
    - **tdbm2ldif** – TDBM and schema unload utility
    - **ldif2tdbm** – TDBM bulkload utility
  - IBM TDS
    - crypt, SHA, MD5, DES, and AES encryption supported for **userPassword** attribute values with the **pwEncryption** configuration option
    - DES and AES encryption of **secretKey** values with the new **secretEncryption** configuration option
    - NULL-based search supported – allows searching LDBM and TDBM backends (SDBM does not participate)
    - **dsconfig** – LDAP configuration utility
    - **ds2ldif** – TDBM, LDBM, and schema unload utility
    - **ldif2ds** – TDBM bulkload utility, schema must be modified beforehand

The IBM TDS supports encryption using AES. It also supports both AES and DES encryption without requiring ICSF. The server also allows encryption of the values of the **secretKey** and **replicaCredentials** attributes in addition to the **userPassword** attribute.

The IBM TDS supports null-based subtree search, which searches all the entries in the TDBM and LDBM backends in a single search. SDBM is not included in null-based subtree searches.

dsconfig is the configuration utility for the IBM TDS server. Its invocation and usage is similar to the **ldapcnf** configuration utility for the ISS LDAP server. The input files and some output members have been renamed to avoid using outdated files. **dsconfig** supports configuring all the IBM TDS backends and all the new configuration options. As with **ldapcnf**, it does not support creating multiple LDBM or TDBM backends.

The **ds2ldif** utility is the new unload utility for the LDBM, TDBM, and schema backends available in IBM TDS. The **tdbm2ldif** utility in the ISS LDAP server only supported the unloading of the TDBM and the schema entry within the TDBM backend(s) configured on the LDAP server.

The **ldif2ds** utility is the new TDBM bulkload utility in IBM TDS. It replaces the **ldif2tdbm** utility in the ISS LDAP server. Note, that in the IBM TDS the schema entry must be modified via an **ldapmodify** command prior to running **ldif2ds**.

Differences Between IBM TDS and ISS - Continued

- Replication Support differences
  - ISS LDAP server
    - All replication for all backends (TDBM only) are controlled by the server as a whole
    - Failing operations can be set aside (APAR OA14456)
  - IBM TDS
    - Supported in a sysplex – uses XCF services to coordinate replication of updates
    - Controlled by each backend (TDBM and LDBM)
      - Individual backend can be a master server, peer server, read-only replica, or not participate in replication
      - Determined by **masterServerDN** or **peerServerDN** configuration options for backend
      - Replica entry in backend directory
    - Failing operations can be set aside

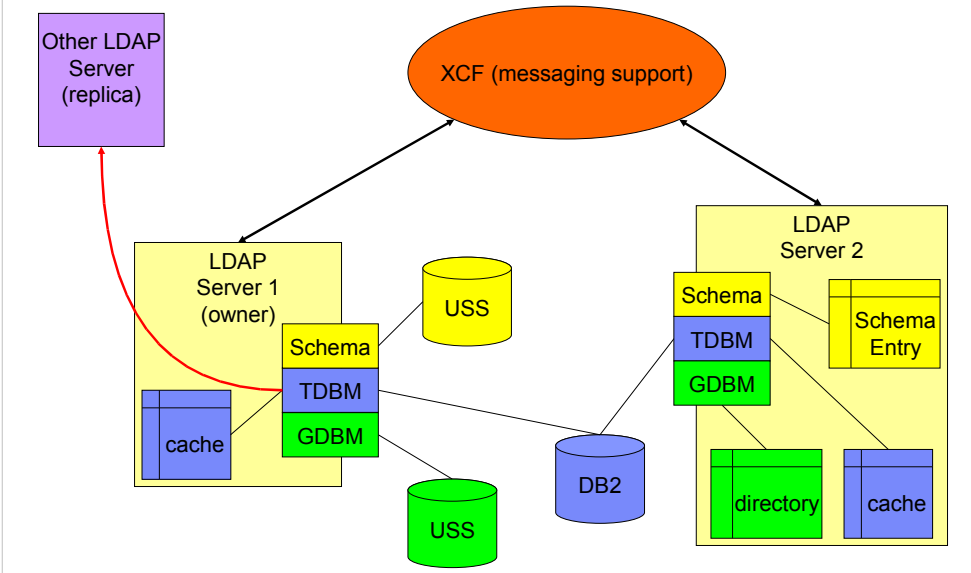
There are several enhancements to replication in IBM TDS. An IBM TDS running in a sysplex can now replicate to an LDAP server outside the sysplex or it can be a replica for an LDAP server outside the sysplex. Replication is now controlled at a backend level. Each LDBM and TDBM backend can individually decide in what way it will participate in replication (or not participate at all). If there is a problem replicating changes to a replica, they can now be set aside in the ISS LDAP server by applying APAR OA14456.

Differences Between IBM TDS and ISS - Continued

- Sysplex Support differences
  - ISS LDAP server
    - Search cache support is disabled in multiserver mode
    - Only used DB2 datasharing – XCF messaging not used
    - LDAP replication only allowed as a read-only consumer/replica
  - IBM TDS
    - Equivalent cache support in single and multiserver mode
    - Event notification via XCF messaging
    - LDAP replication allowed as a read/write master or peer-to-peer in a sysplex

Further details on the IBM TDS sysplex support are on the following slides.

IBM TDS Sysplex Support



USS files must be shared across the sysplex.

The coupling facility (XCF) communicates the updates that are made to either LDAP server 1 or 2 so that the other server knows about the updates. (Note: LDAP only uses XCF messaging for its sysplex support. LDAP does not store data within the XCF.)

As the picture indicates, each LDAP server keeps its own copy of the caches for TDBM. Each server has access to the TDBM backend via DB2 data sharing Since LDAP server 1 is the owner of the sysplex group, LDAP server 2 has an in-memory copy of the GDBM backend and the schema entry. While server 1 is able to write to the schema and GDBM backend files. If a schema or GDBM changelog update is made to server 2, it alerts server 1 about the update via XCF messaging support. For example, if a schema modify operation is targeted against server 2, server 2 will update its in-memory copy of the schema and then alert server 1 about the schema modify update by using XCF messaging. Server 1 will then update the schema file out in USS. If the same update is made to server 1, it will update the schema or GDBM backend and then send the update to server 2 via XCF messaging support.

As mentioned earlier in this presentation, IBM TDS can now be configured as a read/write master or peer-to-peer server. In the ISS LDAP server, LDAP replication from a sysplex group was not allowed.

IBM TDS Sysplex Support - Continued

- First server in sysplex is the owner - Owns the resources
  - USS files (Schema, LDBM, and file-based GDBM)
    - Performs update and sends change to sysplex replicas
    - USS files must be in either zFS or shared HFS and available to all servers in the sysplex
  - Owns LDAP replication out of the sysplex
- Other servers are sysplex replicas
  - LDBM request directory from owner at startup
  - Applies changes from owner
  - Passes modification requests to owner

IBM TDS Sysplex Support - Continued

- TDBM and GDBM (DB2-based) backends are shared between an sysplex replicas
  - Uses DB2 data sharing
  - Notifies other LDAP servers of changes
  - Invalidates internal caches when notified of changes
- When owning server terminates
  - Oldest sysplex replica becomes owner
  - Owns resources
  - Replicates

The oldest sysplex replica is the LDAP server that came up after the sysplex owner.

Migration and Coexistence Considerations

- **Migration**:
  - Can migrate TDBM data in ISS LDAP server to a TDBM or LDBM backend in IBM TDS
  - New messages and reason codes in IBM TDS
  - New operator command syntax in IBM TDS
    - f name,type [options]
- **Coexistence**:
  - Share TDBM data between ISS LDAP server and IBM TDS
    - Some restrictions on both the ISS and IBM TDS
    - Requires coexistence PTF UA25909 on ISS LDAP server

TDBM data in a ISS LDAP server can be migrated to either a TDBM or an LDBM backend in the IBM TDS. In addition, a TDBM database can be shared between an IBM TDS and an ISS LDAP server.

In either case, the differences between the ISS and IBM TDS servers must be understood to ensure that applications use LDAP can operate with the new IBM TDS. Both LDAP servers adhere to the LDAP Version 3 protocol, but this protocol does not specify what messages, reason codes, and, in most instances, return codes must be used. Note that the messages and reason codes are not an API and should not used by an application for programming purposes.

The ISS and IBM TDS servers can be configured to share a TDBM database. There are some restrictions, mainly on the usage of new features of the IBM TDS that are not supported by the ISS LDAP server. The PTF, UA25900, for APAR OA15948 must be applied to the ISS LDAP server to run in this environment.

Migration Considerations

- Configuration file changed
  - Example: DLL names and sysplex configuration options
  - Can use **dsconfig** to create new files and server proc
- TDBM schema migration
  - Done automatically by IBM TDS during initialization
  - Problem if multiple TDBM backends in ISS server with conflicting schema definitions

The ISS LDAP server configuration file will not be acceptable to the IBM TDS. One major change is that all the DLL names are different. For example, the TDBM DLL is GLDBTDBM in the ISS LDAP server configuration file but is GLDBTD31 in the IBM TDS configuration file. A second important change is that the **sysplexServerName** and **sysplexGroupName** options used in the ISS LDAP server configuration file to identify the sysplex are replaced by the **serverSysplexGroup** option in the IBM TDS configuration file.

In the ISS LDAP server, the schema is a separate entry in each TDBM backend. In the IBM TDS, a single global schema is shared for all the backends in the server. The contents of the TDBM backend schema in the ISS LDAP server are migrated automatically to the global schema in the IBM TDS. The IBM TDS does not start if there is an attribute or object class used in a TDBM entry that is not contained in the schema. This can occur on the ISS LDAP server if an attribute or object class is added to the schema, used in an entry, and then removed from the schema. The schema needs to be fixed before the TDBM entry can be migrated. Another schema migration problem can occur if the TDBM backend contains an attribute that uses the boolean syntax with any of these matching rules: caseExactIA5Match, caseIgnoreIA5Match, caseExactOrderingMatch, caseIgnoreOrderingMatch, caseExactSubstringsMatch, or caseIgnoreSubstringsMatch. These matching rules must be removed before the schema can be migrated.

Migration Considerations - Continued

- Replication queue cannot be migrated
  - Make sure ISS LDAP server replication is completed before migration
- Replication in IBM TDS is controlled by each backend
  - May need to add additional replica entries after migration if migrating multiple TDBM backends

The IBM TDS TDBM backend uses different replication tables than in the ISS LDAP server. The contents of the ISS LDAP server replication tables cannot be migrated. Instead, all replication processing must be completed before migration so that these tables are no longer needed. If replication processing is not completed, then some replicas may be out of synch with the master LDAP server after migration.

In the ISS LDAP server, replication entries in one TDBM backend are applied to all TDBM backends. In the IBM TDS LDAP server, each TDBM backend must contain its own replica entries. These entries will need to be added if they were not in the TDBM backend on the ISS LDAP server.

Migration Considerations - Continued

- Migrating ISS LDAP TDBM to IBM TDS TDBM
  - Add TDBM options to IBM TDS server configuration file
  - Run TDBMMGRT SPUFI script to update ISS TDBM tables to IBM TDS TDBM level
    - Creates new replication tables
    - Optionally updates DB_VERSION to '4.0'
      - Do not do if sharing TDBM with ISS LDAP server
  - Start IBM TDS
    - IBM TDS LDAP server merges TDBM backend schema into server global schema
  - Uses TDBM data entries without any migration

The main concern in migrating from a TDBM backend in an ISS LDAP server to a TDBM backend in an IBM TDS is to update the TDBM database so that it contains all the new elements used by the IBM TDS. A SPUFI script file is supplied to add the new database tables and columns. It also updates the DB_VERSION value to '4.0'. This enables the TDBM backend to use the enhanced sysplex and replication support in the IBM TDS. However, an ISS LDAP server cannot use a TDBM database at this level, hence do not update the DB_VERSION if you plan to share the TDBM backend with an ISS LDAP server.

The TDBM entries do not have to be unloaded and reloaded – the IBM TDS can use them as they are. Also, the TDBM backend schema does not have to be manually migrated. The first time the IBM TDS is started with the TDBM backend, it will detect that there is no value for the SCHEMA_TIMESTAMP column and will merge the contents of the TDBM backend schema entry into the IBM TDS server global schema. It then sets the SCHEMA_TIMESTAMP value so that the merge does not occur again.

Migration Considerations - Continued

- Migrating ISS LDAP TDBM to IBM TDS LDBM
  - Add LDBM options to IBM TDS server configuration file
  - Unload schema from ISS LDAP TDBM backend and load into IBM TDS global schema
    - Use **tdbm2ldif** to unload, **ldapmodify** to load
    - Need to edit unload output to remove suffix in DN
      - Note – do not use editor that removes blanks
  - Unload entries from ISS LDAP TDBM backend and load into IBM TDS LDBM backend
    - Use **tdbm2ldif** to unload, **ldapadd** to load

When migrating from a TDBM backend in an ISS LDAP server to an LDBM backend in an IBM TDS, both the TDBM backend schema and the TDBM entries must be unloaded from the ISS LDAP server and then loaded into the IBM TDS. The unloaded schema will start with dn: cn=schema,<suffix>. This must be changed to just dn: cn=schema. Do not use an editor that removes trailing blanks from a line – this could mess up continuation lines in the file and cause load to fail.

The ISS LDAP server unload utility, **tdbm2ldif**, can be used to unload the TDBM schema and entries. Note that the IBM TDS load utility, **ldif2ds**, cannot be used to load an LDBM database – use **ldapadd** instead.

Migration Considerations - Continued

- Migrating ISS LDAP GDBM to IBM TDS DB2-based GDBM
  - Add DB2-based GDBM options to IBM TDS configuration file
  - No LDAP unload – load required – can use data as is
- Converting ISS LDAP GDBM to IBM TDS file-based GDBM
  - Ensure all entries in ISS LDAP server changelog have been processed before conversion
  - Add file-based GDBM options to IBM TDS configuration file

A GDBM backend in an ISS LDAP server can easily be migrated to a DB2-based GDBM backend in an IBM TDS. The only change is to add the backend to the IBM TDS LDAP server configuration file. The GDBM database does not need to be updated to be accessible to the IBM TDS.

A GDBM backend cannot be migrated to a file-based GDBM backend in an IBM TDS. Instead, the changelog entries must first all be processed from the ISS LDAP server.

Sharing Data between IBM TDS and ISS LDAP Server

- ISS LDAP V1R6 or later and IBM TDS can share TDBM and GDBM (DB2-based) databases
  - Configure sysplex and multiserver on each server
  - DB_VERSION must be '3.0' (not '4.0')
  - Function limited to ISS server
    - Enhanced sysplex support and replication disabled
    - Schema updates that contain Non-numeric OID rejected
  - ISS LDAP requires Coexistence PTF UA25909

- ISS LDAP server restrictions
  - All schema updates must be to IBM TDS global schema
    - IBM TDS LDAP server pushes schema updates to TDBM backend schema

A TDBM database can be shared between an ISS LDAP server and an IBM TDS in a sysplex. The ISS LDAP server must be z/OS V1R6 or later (not V1R5). Each server can read and write TDBM entries. Both servers must be configured in multiserver mode with sysplex support. The database level must be '3.0', which is supported by both servers. The PTF for APAR OA15948 must be applied to the ISS LDAP server to allow it to share TDBM data.

There are some restrictions on functionality that can be used when running in this environment. Most of the restrictions are on the IBM TDS to prevent it from using new features that are not supported by the ISS LDAP server. However there is a restriction on the ISS LDAP server that rejects a modification to the TDBM backend schema:

modifying entry cn=schema,cn=ca

ldap_modify: Directory server is unwilling to perform the operation

ldap_modify: additional info: R004175 When a z/OS R6 LDAP level server is sharing a TDBM database with a
z/OS R8 or higher level server, schema modifications must be targeted to the z/OS R8 or higher level server.

All schema changes must be made to the global schema on the IBM TDS. As part of updating the global schema, the IBM TDS also updates the schema entry within the TDBM backend. This makes the schema updates accessible to the ISS LDAP server.

Session Summary

- Overview of the ISS LDAP server and the IBM TDS
- Explained packaging of the LDAP servers within z/OS
- Differences between the ISS LDAP server and the IBM TDS
- Identified migration and coexistence considerations

Publications References

- 'New' IBM TDS
  - SC23-5191 IBM Tivoli Directory Server Server Administration and Use for z/OS
    - Available on z/OS V1R8 as of March 31, 2007
- 'New' IBM TDS LDAP client
  - SA23-2214 IBM Tivoli Directory Server Client Programming for z/OS
    - Available at z/OS V1R8 GA
- 'Old' Integrated Security Services LDAP server
  - SC24-5923 z/OS Integrated Security Services LDAP Server Administration and Use
    - Available at z/OS V1R8 GA