Hands-on Lab:
## Setting up the z/OS LDAP Server with the dsconfig utility.

Background:

The z/OS LDAP server was introduced several years ago. It was a standard LDAP v3 server with support for LDAP v2 if needed. This LDAP server had two (2) primary backend data sources that could be used with it. One was the RACF database called the SDBM in z/OS LDAP terms. The other was the DB2 database(s) called the TDBM. Over the years another DB2 backend was added to the z/OS LDAP server for the change log called the GDBM. The most common use of the z/OS LDAP server usually had a combination of these backend databases.

With z/OS v1.8 there have been some changes to the z/OS LDAP server. All of the past backend databases that existed before z/OS 1.8 are still in existence and are still being used. One of the major changes is the introduction of a new backend data store – a Unix flat file, called LDBM. This new z/OS LDAP server is called the Tivoli Directory Server (TDS) for z/OS.

Because there are several options that can be used to set up the TDS LDAP server, a new configuration utility has been developed. It is called the dsconfig (directory server configuration utility). This utility allows the z/OS TDS server to be configured faster and easier than in previous releases. Basically the dsconfig utility will take parameters as input and create several different JCL jobs that are isolated by functional areas, such as TCP/IP or RACF. The figure below depicts the flow of the dsconfig utility.

The input to the dsconfig utility is a file called ds.profile. Some of the parameters in the ds.profile are pointers to other input files that are advanced detail information about DB2 (ds.db2.profile), RACF (ds.racf.profile), and the z/OS TDS server (ds.slapd.profile). Other parameters within the ds.profile file indicate where system elements are that the z/OS TDS server needs. Also it will indicate an output dataset which will be allocated and populated for the JCL members needed for implementation. Then the appropriate members are reviewed, adjusted if needed, and run by the appropriate people. When all the jobs have been set up, then the z/OS TDS server is ready to be started.

1
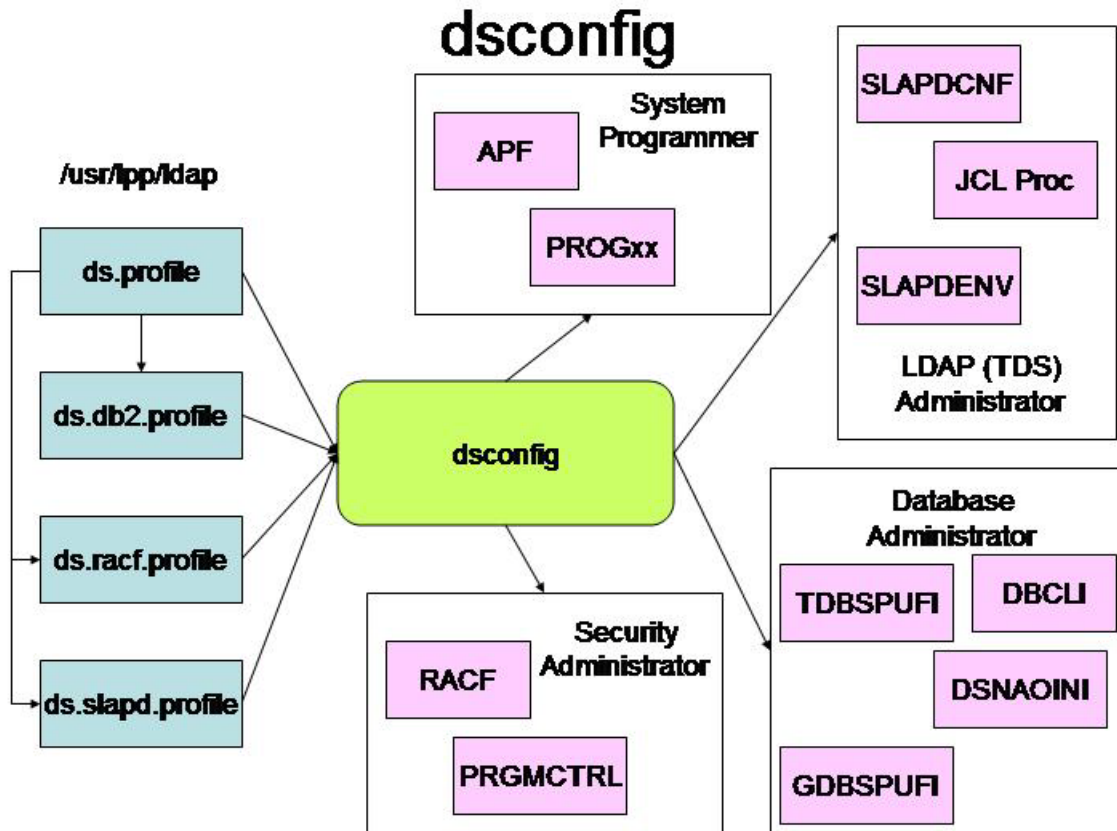
Once the appropriate jobs (members in the output dataset) are adjusted and run, then the z/OS TDS server will be ready to be started, configured, and loaded with data. The z/OS TDS server's infrastructure will look like the picture below. The SLAPDCNF and SLAPDENV files tell the z/OS TDS server how it is to be configured: Which databases will it use (RACF, DB2, etc.)? Will the LDAP server use Kerberos? What is the TCPIP address and port(s)? Will the TDS server support SSL? If it is using SSL, where are the required digital certificates? These files are required and, in this lab, will be in your output dataset from the dsconfig utility. The Keyfile is the location of your digital certificates to allow the z/OS TDS sever to support SSL. This file will be created later in the lab. The two backends that are shown in the picture are the schema and the change log. The schema is required for a generalized TDS server (that is, to support anything other than the RACF database) and it will use a USS file to store the schema definitions that are provided by IBM as part of the z/OS TDS server. The schema definitions supplied by IBM are in the /usr/lpp/ldap directory. We will load the schema into the USS file during the lab. The last file is the optional change log or GDBM. This can be either a USS file or a DB2 database. We will create a GDBM (change log) during the lab and use a USS file for it.

z/OS TDS Infrastructure
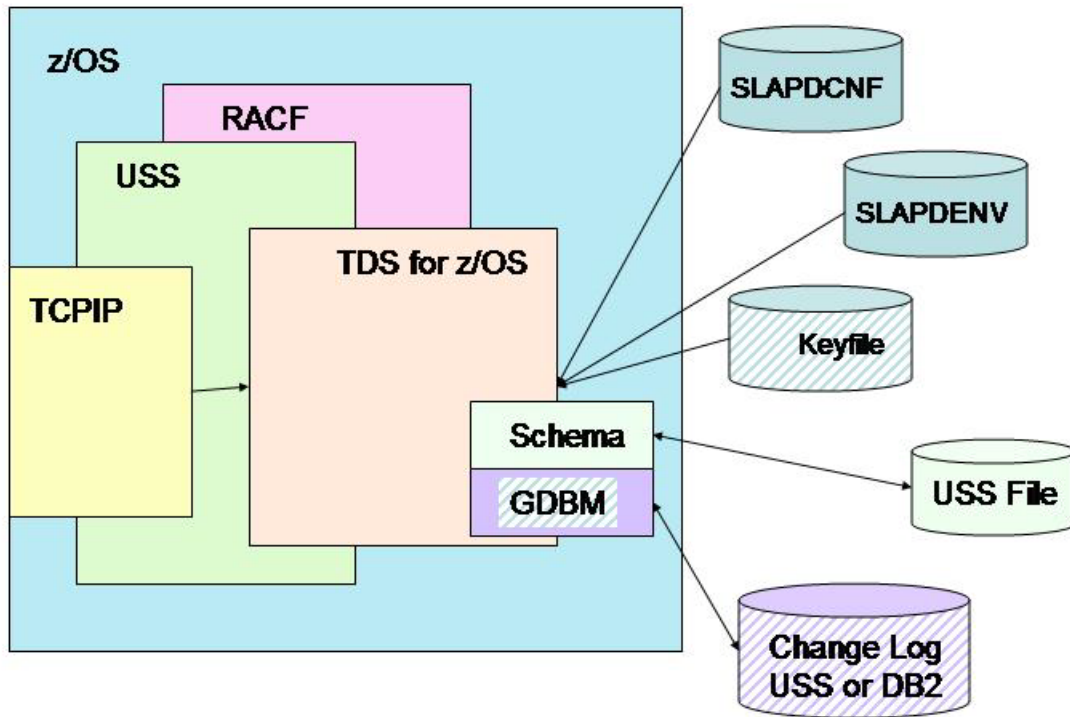
The z/OS TDS server is built with information in the configuration file (SLAPDCNF), the environment variables file (SLAPDENV), and the schema definitions. The keyfile will hold any digital certificates and public/private keys for the z/OS TDS server. And the GDBM is the change log for the z/OS TDS server.

When we are finished with the lab, your z/OS TDS sever will be using data from three (3) databases, plus the schema file. It's backend structure will look like the picture below. The z/OS TDS server will know which dataset to use by the DN (Distinguished Name) in the query and the suffix in the configuration file (SLAPDCNF). There will be schema file which will be identified by the 'cn=schema' suffix. The RACF database will be identified because we will set up the SDBM with a suffix of 'racfdb=local'. The change log (GDBM) will be configured with a suffix of 'cn=changelog' and it will use a USS file for its data. Lastly the TDBM will be our generalized database holding information about some people and organizations. It will be held in a DB2 database with a suffix of 'o=sharedb2'. When it is all configured it will look like this:

3

z/OS TDS Backends
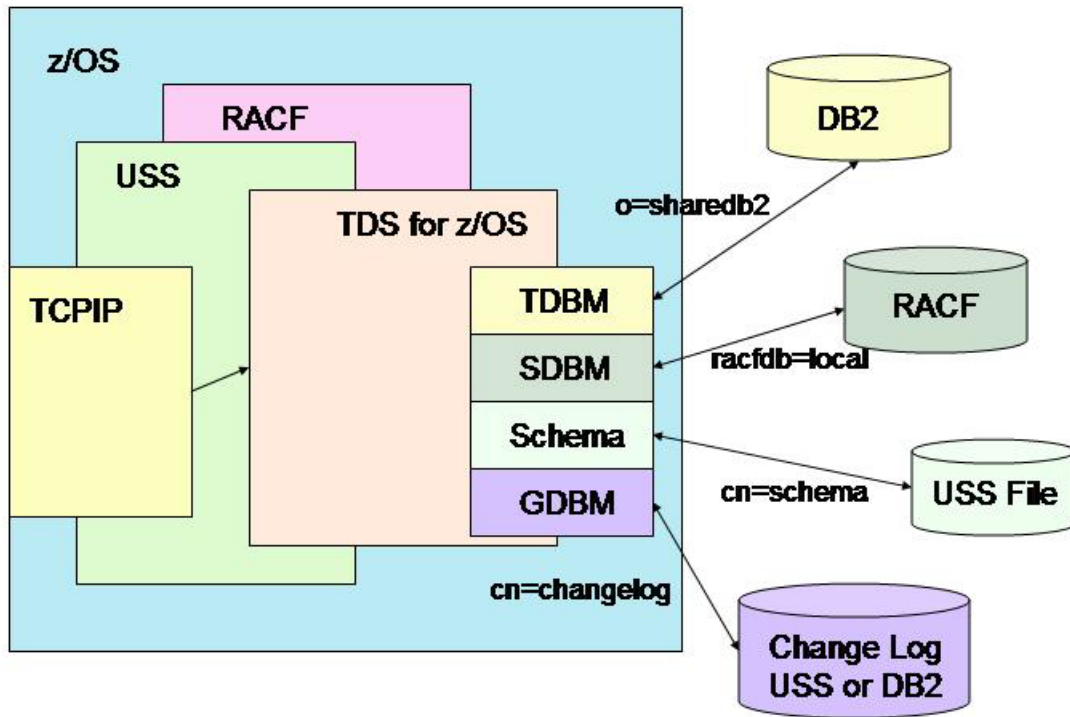
Notes for labs:
Everybody will be sharing the same z/OS system but you will have your individual z/OS TDS server that you can destroy to your heart's content. This is not a typing class or a class on ISPF, SDSF, etc. If you have questions, please ask.

Userids are SHARyxx where y is which lab room (A, B, or C) and xx is your id number (01, 02, … 18, 19, or 20). The name for your z/OS TDS server will be LDAPTMxx, where 'xx' matches your id number. So if you are assigned '01' in room A then your userid will be SHARA01 and your server name will be LDAPTM01.

The IP address that we are using is: mvs1.centers.ihost.com or 32.97.185.7. Your IP ports will depend on both your lab room letter and your id number so if we are in lab room:
- A then our non-SSL port numbers start at 8001 and go to 8020 and the SSL port numbers go from 9501 to 9520. So if your userid is SHARA01 then your ports will be 8001 and 9501 . For SHARA02, they would use ports 8002 and 9502, etc.
- B then our non-SSL port numbers start at 8051 and go to 8070 and the SSL port numbers go from 9531 to 9550. So if your userid is SHARB01 then your ports will be 8051 and 9531. For SHARB02, they would use ports 8052 and 9532, etc.
- C then our non-SSL port numbers start at 8071 and go to 8090 and the SSL port numbers for from 9541 to 9560. So if your userid is SHARC01 then your ports will be 8071 and 9541. For SHARC02, they would use ports 8072 and 9542, etc.

4

Lab 1 - Exercise Instructions:

1. Logon to the MVS system with the information provided by the lab instructor(s).

2. Get into OMVS. You will be in your home directory.

Phase 1 – Review and edit the profiles for the configuration utility:

3. There will be a subdirectory that will contain all the files that you will need to use the **dsconfig** utility and load your z/OS LDAP Server with data. The subdirectory is called **tdslab/tdsconfig**. To get into the subdirectory issue the Unix command:

               cd tdslab/tdsconfig

4. To insure that all the files are there for you, issue the following Unix command: **ls**
The output should include the following files:
ds.db2.profile
ds.profile
ds.racf.profile
ds.slapd.profile
initschema
initsuffixadmin
listadmin
listall
listchange
listroot
listschema
loadsampledata
sample.ldif
schema.IBM.ldif
schema.user.ldif
sdbmivp
suffix_admin.ldif
tdbmivp
If any of these files are missing, please notify your instructor(s).

5. Get into edit on the ds.profile (**oedit ds.profile**). This file has been copied from the /usr/lpp/ldap/etc directory and all the comments have been removed to make it readable. Some of the parameters have changed to point to the appropriate HLQs and subsystems for you. Review the information - several items to note are the OUTPUT_DATASET, LDAPUSRID, TDBM_SUFFIX, SDBM_SUFFIX, ADMINDN, ADMINPW, and the pointers to the advanced profile files at the very bottom of the ds.profile file.

6. Change all the '??' to your userid number. Change all the '!' to your room letter in lower case. Change all the '%' to you room letter in upper case. Before you save and exit from this file, check the OUTPUT_DATASET parameter and all the JOB statements.

5

Make sure all the letters in the dataset names are capital letters. Then save and exit the **ds.profile** file.

7. No changes are required for the **ds.racf.profile** but you might want to review if desired.

8. Edit the **ds.db2.profile** file by changing all the '??' to your userid number and all the '%' to your room letter in upper case. In this file make sure that your room number is a capital letter for all changes. Then save and exit the file.

9. Edit the **ds.slapd.profile** by changing all the '??' to your userid number and all the '!' to your room letter in lower case. In this file, your room letter should be a small letter for all changes. Change the '####' to your assigned unsecured port number. Then save and exit the file.

Phase 2 - Run the configuration utility

10. Now the files are ready for use as described above. To run the dsconfig utility, issue the following Unix command from the directory where the ds.profile file resides (in other words, run this command from the /tdslab/tdsconfig subdirectory):
/usr/lpp/ldap/sbin/dsconfig -i ds.profile

11. This may take several minutes to run so be patient. If you have not changed the OUTPUT_DATASET parameter in the ds.profile, then you will see a message that indicates that you are overwriting the output dataset (the one that I have created for you with the appropriate members). Answer 'yes' to overwrite the dataset – if you don't want to overwrite it, answer 'no' and go back to the ds.profile and change the OUTPUT_DATASET parameter to a different dataset with your HLQ. Then rerun step. You should see the message indicating that the utility is complete if everything ran successfully.

12. The output of the dsconfig utility is in the PDS that is indicated in the OUTPUT_DATASET parameter. If you followed instructions, this output PDS should be **<yourHLQ>.TDSTEST.CNTL**. This may be different if you changed the OUTPUT_DATASET parameter in the ds.profile. Issue the exit command to get out of Unix System Services.

13. From ISPF, get into edit on the **<yourHLQ>.TDSTEST.CNTL** dataset. The following is a list of members:
APF
DBCLI
DSCONFIG
DSENVVAR
DSNAOINI
LDAPSRV
PRGMCTRL

6

PROGDS
RACF
TDBSPUFI
Most of these members, such the APF, RACF, PROGDS, and DBCLI, are actions that are executed only once. They have already been done for you. The ones that will be of interest to you are the DSCONFIG, DSENVVAR, LDAPSRV, and TDBSPUFI members.

Phase 3 - Edit the members in TDSTEST.CNTL

14. In ISPF, edit the **TDBSPUFI** member. This member will be used to create the DB2 environment for your TDBM backend At the bottom of the member, there are several GRANT statements. Delete or comment (by placing a '--' in front of GRANT) all but the last GRANT statement (they have already been done for you). On the last GRANT statement, change the very last word to 'PUBLIC'. This will allow everybody including your LDAP Server to use your newly created database. Save this member and exit.

15. In ISPF, review the **DSCONFIG** member. This member will configure your z/OS TDS server. There should be no changes needed for this member.

16. In ISPF, browse the **LDAPSRV**. This member is your z/OS TDS server PROC (the JCL that starts the STC) and will be copied into the PROCLIB. Save the member.

17. Now we have to rename the **LDAPSRV** member. In the member list of the **TDSTEST.CNTL** dataset, put an 'r' on the line beside the **LDAPSRV** member. A new screen will pop up and ask you to enter the new name. Make the new member name **LDAPTM??** where '??' is your userid number.

18. Lastly, in ISPF, copy the **LDAPTM??** (where ?? is your userid number) from the <yourHLQ>.TDSTEST.CNTL dataset into the **'SHARE.USER.PROCLIB'**.

NOTE: BE SURE THAT YOU EXIT ALL THE WAY OUT OF 'SHARE.USER.PROCLIB' SO THAT YOU FREE THE DATASET SO OTHERS CAN USE IT!!!!!

Phase 4 - Create the DB2 database, tablespaces, tables & indexes:

19. Now we need to allocate the DB2 space for the TDBM. To do this we will execute the SPUFI commands in the **TDBSPUFI** member. Go the **ISPF Primary Option Menu** and scroll down the options until your see the **DB2** option. Be sure that the **HLQ** is set to **DBS9**. Once that is done, enter **DB2**. This will put you into SPUFI - you should see **DB2 Primary Option Menu** as the title and in the upper right corner you should see the DB2 subsystem id which is **DBS9**. If this **SSID** is not set to **DBS9**, then enter option **D**. And make the DB2 Name equal to **DBS9**. Press the enter key twice to get back to the DB2 primary option menu.

From here, enter option **1** to execute SPUFI commands. This will place you on the SPUFI menu, here you will enter **'<yourHLQ>.TDSTEST.CNTL(TDBSPUFI)'** as your input dataset name - NOTE THE QUOTES. And enter **DB2OUT** as your output dataset name (no quotes here). Press the **enter key twice** to display your TDBSPUFI member. You are now in edit on your member and have a last chance to review and edit the member. We have already edited the member so exit the member (**PF3**) to go back to the execution option. Now press the **enter key** again to execute the SPUFI commands. In a couple of second you will see the output from your SPUFI commands. Review all of the return codes which should be zero (0). The most important return is the last one return code which is for the DB2 COMMIT - if this is not zero (0), please notify your instructor(s). Press **PF3** three times to get out of SPUFI.

Phase 5 - Start and verify your z/OS TDS server:

20. With the DB2 database, tablespaces, and tables defined, the z/OS TDS server is ready to be started. Go to SDSF and issue the following command to start the LDAP Server (where ?? is your userid number):
<div align="center">

/s ldaptm??

</div>

21. To be sure that your LDAP Server has started correctly, go to the **DA** option within SDSF and select your LDAP Server. Scroll to the bottom of the displayed output, you should not see any error messages but you will see several information messages describing your TDBM, GDBM, and SDBM backends and the connections to them. The last message that you should see is the:
Ready for requests

22. At this point the z/OS TDS server is up and running with a TDBM (DB2), a GDBM (USS file) and SDBM (RACF) but there is no data in the DB2 backend store. This includes the fact that the full schema has not been loaded into the schema file (a USS file) yet. To define your schema and load some basic data into the LDAP directory, we will issue some Unix commands. To do this you need to get into **OMVS**.

23. Once you are in OMVS, get into the dsconfig subdirectory by entering the following command:
<div align="center">

cd tdslab/tdsconfig

</div>

24. To insure that we have connectivity to your z/OS TDS server and that everything is currently set up correctly, edit the **listroot** file. Change the '####' to your assigned unsecured port number. Save and exit the file. This file will display the root information about the z/OS TDS server, such as your suffix (namingcontext), etc. The namingcontext for the TDBM should be 'o=sharedb2'. Issue the following command:
<div align="center">

listroot

</div>

If you do not see a list of item describing your z/OS TDS server or if you see an error message, please get an instructor. If you do see a list of item, note the last items which should be a description of the changes in your change log.

8

Phase 6 - Load the schema into your z/OS TDS server and test it:

25. To load the schema that IBM provides for you, edit the **initschema** file. Change all the '??' to your userid number. Change all the '!' to your room letter. Change the '####' to your assigned unsecured port number. Save and exit the file. This file will load the schema defined in the two ldif files into your z/OS TDS server. Now execute the ldap commands by entering the following command: **initschema**
Note that the schema files; **schema.user.ldif** and **schema.IBM.ldif,** have already been editted for you so that they point to schema suffix of **cn=schema**. You should see two messages indicating that the ldap server has been modified.

26. You should see two messages indicating that the schema is being updated but to insure that your schema has been updated, edit and run the **listschema** file. Change the '####' to your assigned unsecured port number. Save and exit the file. Then issue the **listschema** command to run the ldap commands. This will place a list of the schema in the **schema.outlist** file. Browse the schema.outlist file, if it is not several pages long then please notify your instructor(s).

Phase 7 - Load required data into the LDAP server and test it:

27. Now we are ready to load data into your LDAP Server. The first data that must be loaded into your LDAP Server is the definition of your suffix (o=sharedb2) and your administrator. The definition of the suffix and administrator has already been built for you in the **suffix_admin.ldif** file. (ldif stands for Ldap Data Interchange File) You can browse this file to see what we are going to load but there are no changes required to this file. The commands to load this data is in the **initsuffixadmin** file. Edit this file by changing all the '??' to your userid number and all the '!' to your room letter. Change the '####' with your assigned unsecured port number. Save and exit this file. Then run the ldap commands by issuing the **initsuffixadmin** command. You should see two messages indicating that o=sharedb2 is being added and that cn=admin,o=sharedb2 is being added.

28. To insure that the suffix and admin have been added, there are two commands to display them. Edit the **listadmin** commands by changing the '####' with your assigned unsecured port number. Save and exit the file. Then run the **listadmin** file which will display the information about the administrator that you just defined in the previous step. To display the suffix information, edit the **listall** file. Change '####' with your assigned unsecured port number. Save and exit the file. Then run the **listall** file which will display all the information in the TDBM of your z/OS TDS server. In this case, this should be your administrator and the suffix.

SDBM & TDBM Backends
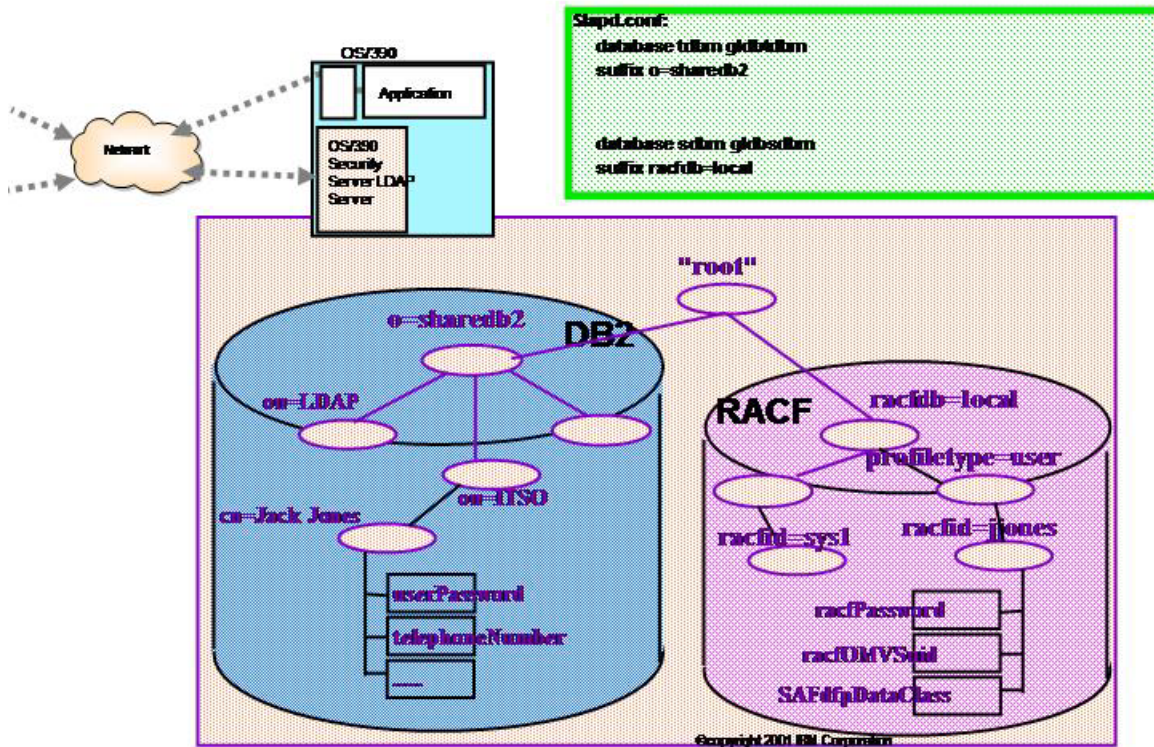
Phase 8 - Load sample data into the z/OS TDS server:

29. Lastly we are going to add some sample data into your ldap server. To do this oedit the **loadsampledata** file. Change all the '??' to your userid number. Change all the '!' to your room letter. Change the '####' to your assigned unsecured port number. Save and exit the file. Then run the **loadsampledata** file which will load all the data that is in the **sample.ldif** file. You should get messages indicating several (actually 51) entries are being added to your LDAP Server.

30. To be sure that the entries have been added to your z/OS TDS server, rerun the **listall** file. Now there should be several more entries displayed - more than just the suffix and administrator entries that were displayed in the above step.

31. At this point, we now have an ldap server that is using the TDBM and SDBM as the backend store with data in both. To display data from either of the backend stores, you can edit and use the **sdbmivp** and **tdbmivp**.
Edit the **tdbmivp** by changing the '####' to your assigned unsecured port number. Save and exit the file. When you run this command, you will list out all the people in your TDBM (DB2) backend and their phone numbers.
Edit the **sdbmivp**. Change all the '??' to your userid number. Change all the '!' to your room letter. Change the '####' to your assigned unsecured port number. Note that to use the SDBM (RACF), you must authenticate with a RACF userid - therefore you will have

10

to change the '%%%%%%%' to your RACF password to get this to run correctly. Save and exit this file. When you execute the **sdbmivp**, you will list out all the RACF information about yourself.

Phase 9 - Listing information in the change log:

32. Now run the **listroot** script again. The last item in the list that is presented back to you is the changelog numbers. This number is different that when you ran the **listroot** command before. This is because of all the add and modify commands that you have been executing against your z/OS TDS server. To see who has been making the changes to your z/OS TDS server, we will list the change log. Edit the **listchange** file. Change the '####' to your unsecured port number. Save and exit the file. Now run the **listchange** script to get a list of all your changes, the changenumber, what was changed, and who did the change.

Now you have built a z/OS TDS server with 3 backends; RACF (SDBM), DB2 (TDBM), and a USS file (GDBM). And you have listed data from each one. The next lab exercises will work with this z/OS TDS server.

Hands-on Lab: Optional Lab SSL
Setting up the RACF digital certificate environment for the z/OS TDS server.

Background:

The TDS server for z/OS can use an SSL connection to protect the integrity of the LDAP data that passes through the network. You can also write client (application) code that uses an SSL connection to a z/OS TDS server. The client code can run on z/OS or distributed. To set up the SSL environment, a digital certificate environment must be set up for the z/OS TDS Server, and, if needed, for the z/OS LDAP client. This digital certificate environment can be set up using either RACF or the Unix utility, gskkyman or a combination of the two facilities. The components that are required for a digital certificate environment are:

1   A public/private key set
2   A signing certificate authority (this must be trusted by both the server and the client)
3   A signed certificate request that is marked as the trusted default digital certificate (implying that there is a certificate request)
4   A keyring (or key database) to store the components mentioned above

The above components must be set up for the z/OS TDS server for server authentication. They must also be setup for the z/OS LDAP client is client authentication is needed. This exercise will set up SSL server authentication with RACF digital certificates in part 1 of the lab. Then in part 2 of the lab, the SSL set up will be expanded to include client authentication. The RACF RACDCERT command will be used to build the environment (via REXX execs).

12

Below is a picture of the final z/OS TDS server setup after the completion of Part 1.
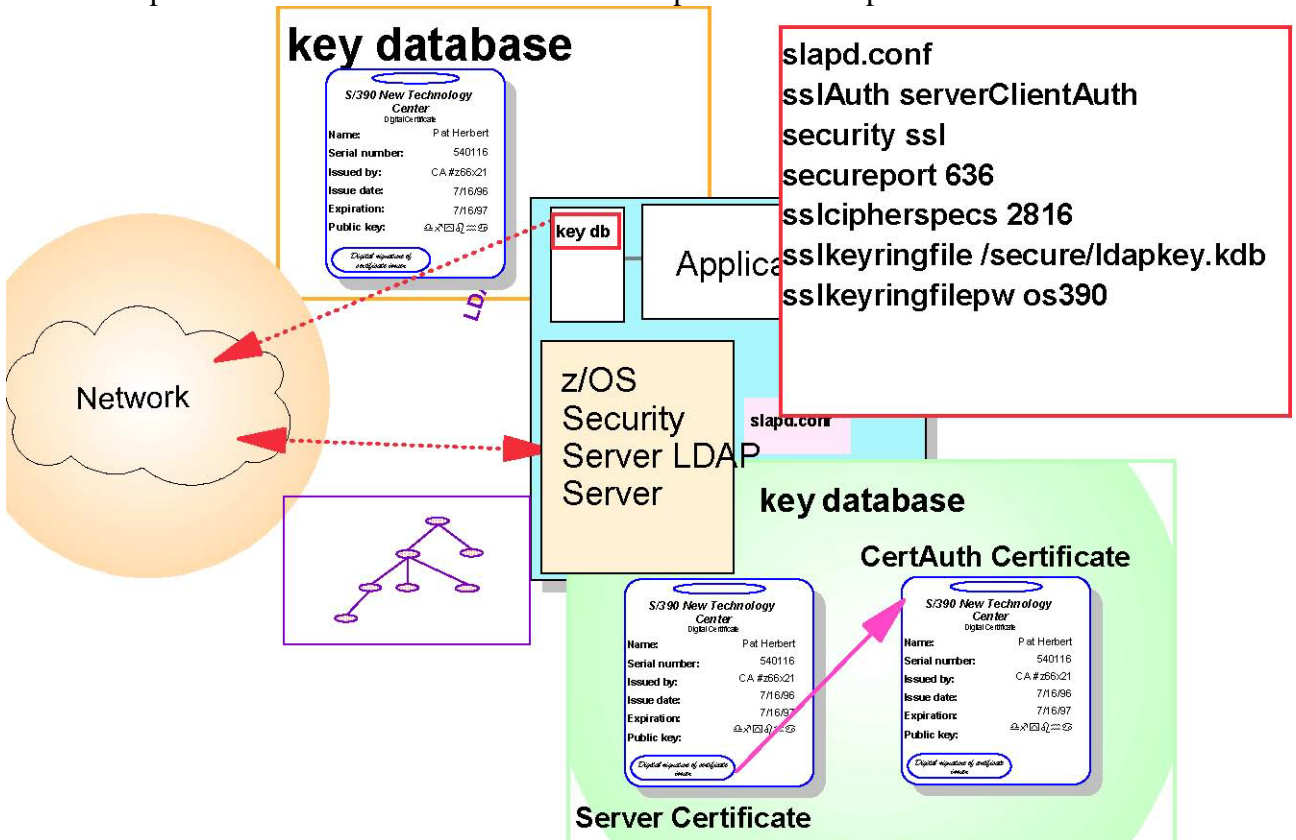
Exercise Instructions (Part 1):
Phase 1 - Verify that your z/OS TDS server is running correctly (you can bypass this phase if appropriate):

1. If you are not already logged onto the MVS system, logon to the MVS system with the information provided by the lab instructor(s).

2. Get into SDSF in DA option. Check to see if your z/OS TDS server is running. Your LDAP Server's STC name will be provided by the lab instructor.

3. In ISPF, check to be sure that **<yourHLQ>.TDSCERT.CLIST** dataset exists. This dataset contains several REXX execs that you will modify and execute to build the SSL environment.

4. In OMVS, check to be sure that there is a subdirectory in your home directory called **tdslab/tdsdcert**. This contains several Unix scripts that execute LDAP commands to test your SSL connection for both server and client authentication.

5. If any of these files do not exist or if your z/OS TDS server is not started, please notify your instructor(s).

6. From OMVS, get into your tdslab/tdsdcert subdirectory by entering the following Unix command (from your home directory):

cd tdslab/tdsdcert

7. Edit the file called **tdbmivp** by entering the following command: **oedit tdbmivp**

8. Change '??' to your team number, change the '!' to your room number, and save the file by pressing the PF3 key.

9. Run the **tdbmivp** file. This will display data from the TDBM (the DB2 database).

10. Edit the file called **sdbmivp**. Change the '??' to your team number, change the '!' to your room number, change the '%%%%%%%' to your RACF password and save the file.

11. Run the **sdbmivp** file. This will display data from the SDBM (the RACF database).

Phase 2 - Stop your LDAP server so you can update its environment:

12. Both of these display commands have presented back to you (the LDAP client) in clear text format. That is, as the data was transported across the network all the data was in the clear and not encrypted via SSL. We are going to set up your LDAP Server to support SSL now. First we must stop the z/OS TDS server so we can change the configuration. Get into SDSF and stop your z/OS TDS Server by entering the following command.
/p <yourLDAPServername>

Phase 3 - Create the Trusted Root digital certificate:

13. In ISPF, edit the **STEP1** member of the **<yourHLQ>.TDSCERT.CLIST** dataset. Change the '??' to your team number, change the '!' to your room number (as a capital letter, please), and save the member by pressing PF3. This REXX exec will issue 2 RACF commands. The first RACDCERT command will create a self-signed digital certificate. This self-signed digital certificate will be used to sign other digital certificate requests. In other words, we are making ourselves a CertificateAuthority and generating other digital certificates. The second RACDCERT command will display the newly created self-signed trusted root certificate.

14. Run **STEP1** by entering the following TSO command from the ISPF option 6 panel:

14

Phase 4 - Create the LDAP server's digital cert, 'sign' it with Trusted Root, and mark it as a trusted digital cert:

15. In ISPF, edit the **STEP2** member of the **<yourHLQ>.TDSCERT.CLIST** dataset. Change the '??' to your team number, change the '!' to your room number (as a capital letter, please), and save the member by pressing PF3. This REXX exec will issue 3 RACF commands. The first RACDCERT command will create the z/OS TDS server's digital certificate and sign it with the self-signed trusted root that you created in STEP1. The second RACDCERT command will mark the z/OS TDS server's digital certificate as a trusted digital certificate. And the third RACDCERT command will display the newly created LDAP Server digital certificate.

16. Run **STEP2** by entering the following TSO command from the ISPF option 6 panel:

ex ldapcert(step2)

Phase 5 - Build the LDAP server's keyring and stored the digital certs:

17. In ISPF, edit the **STEP3** member of the **<yourHLQ>.TDSCERT.CLIST** dataset. Change the '??' to your team number, change '!' to your room number (as a capital letter, please), and save the member by pressing PF3. This REXX exec will issue 2 RACF commands. The first RACDCERT command will create the z/OS TDS server's keyring. The second RACDCERT command will display the newly created LDAP Server keyring. Note that at this time there are no digital certificates connected to the keyring.

18. Run **STEP3** by entering the following TSO command from the ISPF option 6 panel:
ex ldapcert(step3)

19. In ISPF, edit the **STEP4** member of the **<yourHLQ>.TDSCERT.CLIST** dataset. Change the '??' to your team number, change the '!' to your room number (as a capital letter, please), and save the member by pressing PF3. This REXX exec will issue 3 RACF commands. The first RACDCERT command will connect the trusted root that we created in STEP1 to the z/OS TDS server's keyring. The second RACDCERT command will connect the z/OS TDS server's digital certificate that we created in STEP2 to the z/OS TDS server's keyring. And the third RACDCERT command will display the z/OS TDS server keyring with the digital certificates that are now connected to it.

20. Run **STEP4** by entering the following TSO command from the ISPF option 6 panel:
ex ldapcert(step4)

Once this has completed then the LDAP Server's digital certificate environment has been built. But to make it useful, the LDAP configuration file must be updated and the LDAP

Client must be updated so that it can trust the CertificateAuthority.

Phase 6 - Create the client's keyring and store the Trusted Root:

21. In ISPF, edit the **STEP5** member of the **<yourHLQ>.TDSCERT.CLIST** dataset. Change the '??' to your team number, change '!' to your room number (as a capital letter, please), and save the member by pressing PF3. This REXX exec will issue 3 RACF commands. The first RACDCERT command will create the LDAP client's keyring. The second RACDCERT command will display the LDAP client's keyring. Note that the keyring is empty. And the third RACDCERT command will display all the keyring associated with this RACF userid (that is the LDAP Client's RACF userid). Note that there is only one RACF keyring associated with the LDAP Client's RACF userid.

22. Run **STEP5** by entering the following TSO command from the ISPF option 6 panel:
ex ldapcert(step5)

23. In ISPF, edit the **STEP6** member of the **<yourHLQ>.TDSCERT.CLIST** dataset. Change the '??' to your team number, change the '!' to your room number (as a capital letter, please), and save the member by pressing PF3. This REXX exec will issue 2 RACF commands. The first RACDCERT command will import (connect) the self-signed trusted digital certificate that was created in STEP1 to the LDAP client's keyring. The second RACDCERT command will display the LDAP client's keyring. Note that the keyring now has the CertificateAuthority's digital certificate in it and that it is marked as TRUSTed.

24. Run **STEP6** by entering the following TSO command from the ISPF option 6 panel:
ex ldapcert(step6)

Phase 7 - Update the LDAP server's SLAPD configuration file to use the server's keyring and restart the LDAP server:

25. Now the z/OS TDS server must be configured to support SSL. In ISPF, edit the **DSCONFIG** member of the **<yourHLQ>.TDSTEST.CNTL** dataset. Now we have to tell the z/OS TDS server which port is our SSL port. Find the last word, 'listen', in the file. You do this with the 'f listen last' command in ISPF edit. The line should look like this: `#listen ldap://:pc`   Delete the '#'. Add an 's' onto the end of ldap. Change the 'pc' to the ssl port that your instructor gave to you. So the line now looks like this: `listen ldaps://:????`

Now find the line in the configuration file that is the start of the SSL definitions – it will look like this: `SSL/TLS specific CONFIGURATION SETTINGS`. You should be able to find it if you do a find on 'SSL'. For the change the following parameters so they look like this (note that the leading '#' must be deleted in these 2 lines):

SSLAuth serverAuth   (delete the '#' from the first column of this line)
sslKeyRingFile RACF_Keyring_for_LDAPTM?? (where ?? is your team number – be careful with the capital letters)

26. Go into SDSF and restart your LDAP Server by entering the following command:

                     /s <yourLDAPServername>

Phase 8 – Test SSL with server authentication only:

27. To check that the z/OS TDS server is running in SSL mode, under SDSF, go into the DA option and select your LDAP server. At the top of the listing, you should find a message that indicates that the LDAP Server has activated a secure communication on a certain port. If this message does not appear then the SSL support in the LDAP Server is not set up correctly.

28. Now we are ready to test our SSL connection with some LDAP client commands. Go into OMVS and get into your **tdslab/tdsdcert** subdirectory.

29. To test if an SSL connection can be established correct, edit the **racf_ssl_sr1** file. Replace the '??' in the file with your team number and the '!' with the capital room letter. Change the port number (the –p parameter) with the SSL port number and save the file.

30. Run the **racf_ssl_sr1** script. This command will set up an SSL connection to the LDAP Server and return all the DNs in the LDAP Server (TDBM).

31. To demonstrate that SSL can be used for RACF information as well as for DB2 data, edit the **sslserver_good** file**.** Change all the '??' to your team number, change the '%%%%%%%' to your RACF password, and change all the '!' to the capital room letter. Change the port number (the –p parameter), change the '####' with the SSL port number and save the file. Then run the **sslserver_good** script. This will set up an SSL connection and then authenticate the LDAP client with a RACF userid before returning RACF information back to the client.

32. Of course, you will never make a mistake, so we have include an example that will show you a common error message. Edit the **sslserver_bad** file. Change the '??' to your team number, change the port number, the '####', to your SSL port number, and save the file. Run the **sslserver_bad**. You should get an error message that indicates that the ldap_ssl_client_init failed with a 113 return code. This basically means that for some reason the SSL connection could not be established because of a keyring access problem. Can you spot the problem?

Phase 9 - Create a client digital cert (using the same Trusted Root) and store it in the client's keyring:

33. Now, if everything is working correctly, we are ready to set the LDAP client up to a digital certificate for authentication. In ISPF, edit the **STEP7** member of the **<yourHLQ>.TDSCERT.CLIST** dataset. Change the '??' to your team number, change the '!' to the capital room letter, and save the member by pressing PF3. This REXX exec will issue 3 RACF commands. The first RACDCERT command will create the LDAP client's digital certificate and sign it with the self-signed trusted root that you created in STEP1. The second RACDCERT command will mark the LDAP client's digital certificate as a trusted digital certificate. And the third RACDCERT command will display the newly created LDAP client digital certificate. Note that this digital certificate is associated with your RACF userid.

34. Run **STEP7** by entering the following TSO command from the ISPF option 6 panel:
ex ldapcert(step7)

35. In ISPF, edit the **STEP8** member of the **<yourHLQ>.TDSCERT.CLIST** dataset. Change the '??' to your team number, change the '!' to the capital room letter, and save the member by pressing PF3. This REXX exec will issue 2 RACF commands. The first RACDCERT command will import (connect) the LDAP client's digital certificate that was created in the previous step to the LDAP client's keyring. The second RACDCERT command will display the LDAP client's keyring. Note that the LDAP Client's digital certificate is marked as the DEFAULT digital certificate.

36. Run **STEP8** by entering the following TSO command from the ISPF option 6 panel:
ex ldapcert(step8)

Phase 10 - Update the LDAP server's SLAPD configuration file to support client authentication with digital certificates and restart the LDAP server:

37. Now the LDAP Server must be configured to support Client Authentication. In ISPF, edit the **DSCONFIG** member of the **<yourHLQ>.TDSTEST.CNTL** dataset. Find the SSL definitions as you did before. Change the **sslAuth** parameter to **serverClientAuth**. Save the **DSCONFIG** member.

38. Go into SDSF and restart your LDAP Server by entering the following command:
/p <yourLDAPServername>

Wait a couple of seconds and then enter the following command**:**

/s <yourLDAPServername>

Phase 11 - Test authenticating your client with your digital cert:

39. Now we are ready to test our SSL connection with some LDAP client commands. Go into OMVS and get into your **tdslab/tdsdcert** subdirectory.

40. To test if an SSL connection can be established correct, edit the **sslclient_good** file. Replace all the '??' in the file with your team number and the '!' with the capital room letter. Change the port number, the '####', to your SSL port number and save the file.

41. Run the **sslclient_good** script. This command will set up an SSL connection and authenticate the LDAP Client with the DN in the client's digital certificate. The output that should be returned to the client is all the DNs in the LDAP Server (TDBM).

42. It is important to remember that if the client is going to use a digital certificate then the LDAP Server will authenticate the client with the DN that is presented in the client's digital certificate. To demonstrate this, edit the **sslclient_bad** file. Change the '??' to your team number. Change the port number, the '####', to your SSL port number and save the file. Run the **sslclient_bad** script. You should get the same error message as before, indicating that there is a problem accessing the keyring.

43. Now edit the **sslclient_bad** file again. Change the word 'NOTHERE' to your RACF userid. Save the file and run. What output did you get? Why didn't it work?

44. At this point, you have got your LDAP Server running with SSL that supports both server and client authentication.

19