**Setup TDE for 19c**

# source DB environment

**Step 1- Create Directory & Set wallet\_root**

mkdir -p $ORACLE\_BASE/admin/DB\_UNIQUE\_NMAE/wallet

sqlplus -s "/as sysdba"

SQL> sho parameter wallet

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

ssl\_wallet string

wallet\_root string

SQL>

ALTER SYSTEM SET WALLET\_ROOT=‘{{ORACLE\_BASE}}/admin/{{db\_unique\_name|upper}}/wallet’ SCOPE=SPFILE;

SQL> alter system set wallet\_root='/oracle/app/oracle/product/admin/AM1UO004/wallet' scope=spfile;

**Step 2- Bounce Database**  
sqlplus -s "/as sysdba"

SQL> SHUTDOWN IMMEDIATE

**Step 3- Backup the Database**

**Step 4- Start Database**

sqlplus -s "/as sysdba"

SQL> STARTUP

**Step 5- Set TDE\_CONFIG to FILE**

ALTER SYSTEM SET TDE\_CONFIGURATION="KEYSTORE\_CONFIGURATION=FILE" SCOPE = BOTH;

SQL> sho parameter wallet

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

ssl\_wallet string

wallet\_root string

SQL> alter system set tde\_configuration="KEYSTORE\_CONFIGURATION=FILE" scope=both;

System altered.

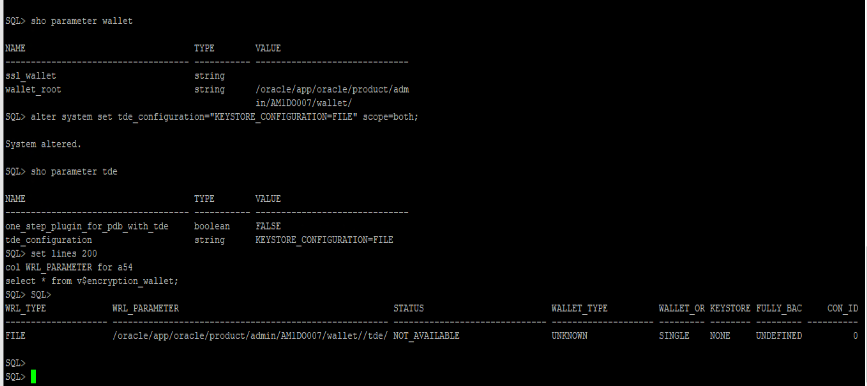
SQL> sho parameter tde

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

one\_step\_plugin\_for\_pdb\_with\_tde boolean FALSE

tde\_configuration string KEYSTORE\_CONFIGURATION=FILEstring



SQL> set lines 200

col WRL\_PARAMETER for a54

select \* from v$encryption\_wallet;

SQL> select t.name,e.ENCRYPTIONALG,e.ENCRYPTEDTS from V$ENCRYPTED\_TABLESPACES e, v$tablespace t where t.ts#=e.ts#(+);

**Step 6- Create Keystore**

ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '{{ oracle\_base }}/admin/{{ db\_unique\_name| upper }}/wallet/tde' IDENTIFIED BY {{ wallet\_pass }};

SQL> ADMINISTER KEY MANAGEMENT CREATE KEYSTORE IDENTIFIED BY Haleon#123;

keystore altered.

SQL> set lines 200

col WRL\_PARAMETER for a54

select \* from v$encryption\_wallet;

SQL> SQL>

WRL\_TYPE WRL\_PARAMETER STATUS WALLET\_TYPE WALLET\_OR KEYSTORE FULLY\_BAC CON\_ID

-------------------- ------------------------------------------------------ ------------------------------ -------------------- --------- -------- --------- ----------

FILE /oracle/app/oracle/product/admin/AM1DO007/wallet//tde/ CLOSED UNKNOWN SINGLE NONE UNDEFINED 0

**Step 7- Open the Keystore**

administer key management set keystore open force keystore identified by {{ wallet\_pass }} ;

SQL> administer key management set keystore open identified by Haleon#123;

keystore altered.

SQL> set lines 200

col WRL\_PARAMETER for a54

select \* from v$encryption\_wallet;

SQL> SQL>

WRL\_TYPE WRL\_PARAMETER STATUS WALLET\_TYPE WALLET\_OR KEYSTORE FULLY\_BAC CON\_ID

-------------------- ------------------------------------------------------ ------------------------------ -------------------- --------- -------- --------- ----------

FILE /oracle/app/oracle/product/admin/AM1DO007/wallet//tde/ OPEN\_NO\_MASTER\_KEY PASSWORD SINGLE NONE UNDEFINED 0

SQL>

**Step 8- Create Master Key / Set master encryption key**

ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY {{ wallet\_pass }} WITH BACKUP ;

SQL> administer key management set key force keystore identified by Haleon#123 with backup;

keystore altered.

SQL> set lines 200

col WRL\_PARAMETER for a54

select \* from v$encryption\_wallet;

SQL> SQL>

WRL\_TYPE WRL\_PARAMETER STATUS WALLET\_TYPE WALLET\_OR KEYSTORE FULLY\_BAC CON\_ID

-------------------- ------------------------------------------------------ ------------------------------ -------------------- --------- -------- --------- ----------

FILE /oracle/app/oracle/product/admin/AM1DO007/wallet//tde/ OPEN PASSWORD SINGLE NONE NO 0

SQL>

**Step 9- Convert keystore to AUTOLOGIN**

ADMINISTER KEY MANAGEMENT CREATE AUTO\_LOGIN KEYSTORE FROM KEYSTORE '{{ oracle\_base }}/admin/{{ db\_unique\_name| upper }}/wallet/tde' IDENTIFIED BY {{ wallet\_pass }};

SQL> administer key management create auto\_login keystore from keystore '/oracle/app/oracle/product/admin/AM1UO004/wallet/tde/' identified by Haleon#123;

keystore altered.

SQL> set lines 200

col WRL\_PARAMETER for a54

select \* from v$encryption\_wallet;

SQL> SQL>

WRL\_TYPE WRL\_PARAMETER STATUS WALLET\_TYPE WALLET\_OR KEYSTORE FULLY\_BAC CON\_ID

-------------------- ------------------------------------------------------ ------------------------------ -------------------- --------- -------- --------- ----------

FILE /oracle/app/oracle/product/admin/AM1DO007/wallet//tde/ OPEN PASSWORD SINGLE NONE NO 0

SQL>

**Step 10- Close Wallet / Close the password Keystore and open the auto-login Keystore**

administer key management set keystore close identified by {{ wallet\_pass }};

SQL> administer key management set keystore close identified by Haleon#123;

keystore altered.

SQL> set lines 200

col WRL\_PARAMETER for a54

select \* from v$encryption\_wallet;

SQL> SQL>

WRL\_TYPE WRL\_PARAMETER STATUS WALLET\_TYPE WALLET\_OR KEYSTORE FULLY\_BAC CON\_ID

-------------------- ------------------------------------------------------ ------------------------------ -------------------- --------- -------- --------- ----------

FILE /oracle/app/oracle/product/admin/AM1DO007/wallet//tde/ OPEN AUTOLOGIN SINGLE NONE NO 0

SQL> select con\_id, wallet\_type,wrl\_parameter, status from v$encryption\_wallet;

####TDE keystore details####

SQL> set linesize 250 pagesize 250

column name format a40

column masterkeyid\_base64 format a60

select name,utl\_raw.cast\_to\_varchar2( utl\_encode.base64\_encode('01'||substr(mkeyid,1,4))) || utl\_raw.cast\_to\_varchar2( utl\_encode.base64\_encode(substr(mkeyid,5,length(mkeyid)))) masterkeyid\_base64 FROM (select t.name, RAWTOHEX(x.mkid) mkeyid from v$tablespace t, x$kcbtek x where t.ts#=x.ts#);

SQL> select con\_id, wallet\_type,wrl\_parameter, status from v$encryption\_wallet;

SQL> select t.name,e.ENCRYPTIONALG,e.ENCRYPTEDTS from V$ENCRYPTED\_TABLESPACES e, v$tablespace t where t.ts#=e.ts#(+);

**Step 11- Encrypt the tablespace**

SQL> select t.name,e.ENCRYPTIONALG,e.ENCRYPTEDTS from V$ENCRYPTED\_TABLESPACES e, v$tablespace t where t.ts#=e.ts#(+);

SQL> Alter tablespace USERS encryption online encrypt;

SQL> Alter tablespace TOOLS encryption online encrypt;

SQL> select t.name,e.ENCRYPTIONALG,e.ENCRYPTEDTS from V$ENCRYPTED\_TABLESPACES e, v$tablespace t where t.ts#=e.ts#(+);

**Step 12- Cronjob for taking Master key backup**

#Every 12hrs wallet backup

00 \*/12 \* \* \* /oracle/dba/wallet\_AM1DO007\_env.sh AM1DO007 < /dev/null

cat /oracle/dba/wallet\_$ORACLE\_SID\_env.sh

#!/bin/ksh

export ORACLE\_HOME=/oracle/app/oracle/product/19.3.0

export ORACLE\_SID=AM1DO007

export PATH=$PATH:$ORACLE\_HOME/bin

cp -rp /oracle/app/oracle/product/admin/AM1DO007/wallet/tde/cwallet.sso /oracle/dba/wallet\_backup/AM1DO007/cwallet.sso\_"$ORACLE\_SID"\_$(date "+%Y.%m.%d-%H.%M.%S")

cp -rp /oracle/app/oracle/product/admin/AM1DO007/wallet/tde/ewallet.p12 /oracle/dba/wallet\_backup/AM1DO007/ewallet.p12\_"$ORACLE\_SID"\_$(date "+%Y.%m.%d-%H.%M.%S")

* *algorithm* is one of the following values to specify AES encryption:
  + AES128
  + AES192
  + AES256

ALTER TABLESPACE SYSTEM ENCRYPTION ONLINE ENCRYPT

FILE\_NAME\_CONVERT=('system01.dbf','system01\_enc.dbf');

ALTER TABLESPACE SYSTEM ENCRYPTION OFFLINE ENCRYPT;

DROP TABLESPACE temp\_01;

CREATE TEMPORARY TABLESPACE temp\_01

TEMPFILE 'temp01.dbf' SIZE 5M AUTOEXTEND ON

ENCRYPTION ENCRYPT;

Q- Can TDE be disabled/removed in the database? Can the TDE wallet be deleted and recreated if the wallet or its password is lost?

Ans- It is not possible to remove TDE once implemented.

Even if there are no encrypted objects in the database, the TDE wallet has to be present in the wallet location.It does not cause any harm.

Also, the metadata related to TDE keys cannot be cleaned up from the dictionary manually.

Deleting the TDE wallet will not disable TDE. Once the TDE wallet is configured, the wallet **should never** be deleted or recreated.

Recreating the wallet using any parameters is **not supported**.

Oracle Support/Development team **will not** help in resolving any issues arising due to such operations.

If the wallet or its password is lost, then there is no way to recover the encrypted data.There is no back door, the database has to be recreated.