SAPtips Journal

Copying and Recreating a Live 4.7 Production System: An Illustrated Guide

By Joey Hirao, JoTech LLC.

Editor's Note: Basis Editor Joey Hirao has shown us some nifty tricks in the last year and a half, but he might have topped himself this time. In this edition of SAPtips, Joey walks readers through the process of recreating a live 4.7 production environment. So why recreate a live system? As Joey has learned, SAP® projects that maintain a live copy have a lot of useful systems flexibility. Perhaps the most important use of a copied SAP system? The ability to test new functions on live SAP data. Sandbox tests are fine as far as they go, but a full live-system copy is a much more powerful testing tool. Using actual coding samples and screen shots, Joey takes readers step by step through the process of recreating a 4.7 SAP system.

Naming Convention	Description
SRC	Source database system identifier
TGT	Target database system identifier
srchost	Source database hostname
tgthost	Target database hostname
SAPR31	Schema name for SRC and TGT

Figure 1: Naming Conventions Used in This Article

Introduction

The ability to quickly and easily recreate SAP systems is an extremely valuable tool. The importance and benefit becomes evident after going live with SAP. There are many uses for a copy of the production instance, including, but not limited to, testing and problem analysis. Plain and simple, there is nothing better than testing on a copy of live data. I have seen installations copy their production SAP database back to the test system as frequently as on a daily basis. This article will describe the technical procedure in copying a 4.7 UNIX and Oracle 9i SAP system.

```
SQL> select * from E070L;

LASTNUM  TRKORR
---------- --------------------
TRKORR          TGTK900010
```

Figure 2: SQL Statement to Obtain Transport

Overview

Since the introduction of 4.7, there have been specific items unique to the new SAP version. In prior releases, the database schema name was SAPR/3. As of 4.7, the schema is defined during install. With that in mind, you can plan your installations with specific schema names. For example, at one customer, I have set up landscapes (development, test, production) with the same schema name. Conversely, at other sites, I have created all unique schema names. If the schema names are different between systems, UNIX environmental variables will need to be modified after a database copy from one system to another.

The copy process can be segmented into three distinct steps:

1. Preparatory phase

2. Restore phase

3. Post copy phase

This article will illustrate each of these steps in detail, using actual scripts and syntax to show how each phase is executed.

To facilitate the discussion, Figure 1 shows the naming conventions used throughout this paper.

Additionally, the example in this article makes the following assumptions:

1. SAP system TGT installed and operational on tgthost.

2. SAP system SRC installed and operational on srchost.

Preparatory Phase

1. Prior to initiating the copy, there may be some steps you would want to take. If you create transports in the TGT system, you will want to keep the transport numbering. Start by recording the transport. Execute a SQL command and retrieve the table contents (see Figure 2).

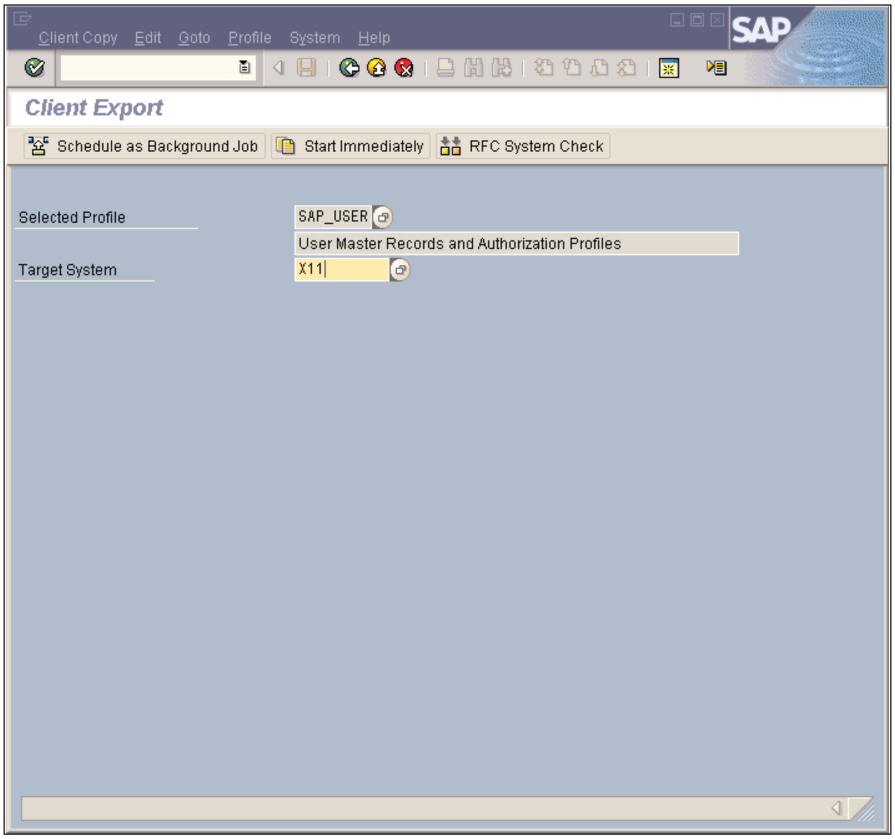
2. If you desire to keep the users from TGT, export the users and authorization via transaction code SCC7, client export. Select the profile SAP_USER to capture user

records and authorization profiles (see Figure 3).

3. Take an offline backup of the SRC database. This will be the backup used for the copy to TGT. You may use SAP's br (backup and restore) tools, operating system backup utilities, or third-party backup tools. Figure 4 shows the syntax for a backup using SAP's br tools interfaced with a third-party backup management software.

4. Create a script to create the controlfiles for TGT. Logon to the SRC database and issue the SQL statement shown in Figure 5.

   The SQL statement will produce a file in the user_dump_dest directory. This file will be used to create the script control.sql, (see Figure 6).



Selected Profile: SAP_USER

User Master Records and Authorization Profiles

Target System: X11

Figure 3: Exporting Users Via Transaction Code SCC7

```
orasrc@srchost% brbackup –u system/password -d util_file -t offline_force -m all
```

Figure 4: Backup SRC

```
SQL> alter database backup controlfile to trace;

Database altered.
```

Figure 5: SQL Statement to Backup controlfile

```
orasrc@srchost% cd /oracle/SRC/saptrace/usertrace

orasrc@srchost% ls -ltr
total 20
-rw-r-----   1 orasrc  dba     1613 Mar 22 13:53 src_ora_1454.trc
-rw-r-----   1 orasrc  dba     7471 Apr 20 11:31 src_ora_4474.trc

orasrc@srchost% cp src_ora_4474.trc /tmp/control.sql
```

Figure 6: Preparing to Create control.sql

5. Edit control.sql and perform the following changes:

   a. Change all occurrences of SRC to TGT.

   b. Change REUSE to SET.

   c. Remove all lines above the second occurrence of STARTUP NOMOUNT.

   d. Refer to control.sql referenced in the Appendix as a sample.

   e. Identify background jobs that need to be scheduled in TGT system.

Restore Phase

Prior to restoring the files onto the TGT system, ensure to stop system TGT. Issue the command stopsap as user tgtadm. The restore of the database files will vary depending on the method used to backup SRC. The following steps describe a restore using SAP's br tools.

1. First, identify the backup set from which to restore. Query the log files to review the latest offline backups. See Figure 7 to identify which backup set to utilize.

2. Ensure that the backup chosen is completed successfully. See Figure 8 for the procedure to check for backup status.

3. Issue the "brrestore" command to restore the datafiles to the TGT database (see Figure 9).

4. After the physical restore is completed, remove the controlfiles for TGT (see Figure 10).

5. Execute the script control.sql that was created during the Preparatory Phase as shown in Figure 11. The script will create the controlfile for TGT, open the database, and add the temporary tablespace for TGT.

```
orasrc@srchost% cd /oracle/SRC/sapbackup
orasrc@srchost% ls –lrt *.aff
-rw-r--r--  1 orasrc  sapsys   24299 Apr 12 23:04 bdnbirgk.aff
-rw-r--r--  1 orasrc  sapsys   24556 Apr 16 01:50 bdnbxkra.aff
-rw-r--r--  1 orasrc  sapsys   24660 Apr 20 01:52 bdncryyz.aff
```

Figure 7: Identify Backup Set

```
orasrc@srchost% tail bdncryyz.aff

BR0280I BRBACKUP time stamp: 2004-04-20 01.52.03
BR0307I Shutting down database instance SRC ...

BR0280I BRBACKUP time stamp: 2004-04-20 01.52.10
BR0308I Shutdown of database instance SRC successful

BR0056I End of database backup: bdncryyz.aff 2004-04-20 01.52.02
BR0280I BRBACKUP time stamp: 2004-04-20 01.52.10
BR0052I BRBACKUP terminated successfully
```

Identify backup log filename. Copy backup log file to target host. In this example, the log file name of the latest backup is bdncryyz.aff.

Figure 8: Check Backup Status Procedure

```
orasrc@srchost% brrestore -m all –b bdncryyz.aff
```

Figure 9: Start brrestore

6. Create the necessary Oracle users by executing script sapuser.sql. See the Appendix for an example script. The sapuser.sql script does the following:

   a. Creates the OPS$ accounts for the new system.

   b. Creates the SAPUSER table.

   c. Drops the SRC OPS$ accounts.

Execute sapuser.sql in the same fashion as in the previous step (step 5). Be advised that if the schema names were different between SRC and TGT, which in this case they are not, the environmental variable dbs_ora_schema needs to be adjusted on the tgthost for users oratgt and tgtadm.

```
        SAPDBA V6.20 - SAP Database Administration
    ______________________________________________________

           ORACLE version: 9.2.0.4.0
           ORACLE_SID    : TGT
           ORACLE_HOME   : /oracle/TGT/920_64
           DATABASE      : open
           SAPR3M                     : 620, 13 times connected

      a - Startup/shutdown instance   h - Backup database
      b - Instance information        i - Backup offline redo logs
      c - Tablespace administration   j - Restore/recovery
      d - Reorganization              k - Additional functions
      e - Export/import               l - Show/cleanup
      f - Archive mode                m - User and security

      q - Quit

      Please select ==> m
```

Figure 12: sapdba tool

7. Initialize SAP connect using SAP's sapdba tool. Start sapdba as user oratgt and select m (User and security) (see Figure 12).

8. Select d (Initialize SAP connect), (see Figure 12a).

```
  oratgt@tgthost% cd /oracle/TGT
  oratgt@tgthost% find . –name "cntrl*.dbf"
  ./origlogA/cntrl/cntrlTGT.dbf
  ./sapdata1/cntrl/cntrlTGT.dbf
  ./saparch/cntrl/cntrlTGT.dbf
  oratgt@tgthost%      rm      ./origlogA/cntrl/cntrlTGT.dbf      ./sapdata1/cntrl/cntrlTGT.dbf      \
./saparch/cntrl/cntrlTGT.dbf
```

Figure 10: Remove controlfiles

```
oratgt@tgthost% sqlplus "/ as sysdba"

SQL*Plus: Release 9.2.0.4.0 - Production on Thu Apr 22 17:34:12 2004

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle9i Enterprise Edition Release 9.2.0.4.0 - 64bit Production
With the Partitioning option
JServer Release 9.2.0.4.0 - Production
SQL> @control.sql
```

Figure 11: Execute SQL Script

Post-Copy Phase

Now that the copy is complete, some data needs to be adjusted in the TGT system in order to properly function as a copied SAP system. Use the following steps.

1. Execute trunc.sql to truncate selected tables. Run the trunk.sql script in the same manner as the previous scripts were executed. Refer to Figure 11 for example syntax. The script trunc.sql deletes all rows of tables that contain data specific to SRC.

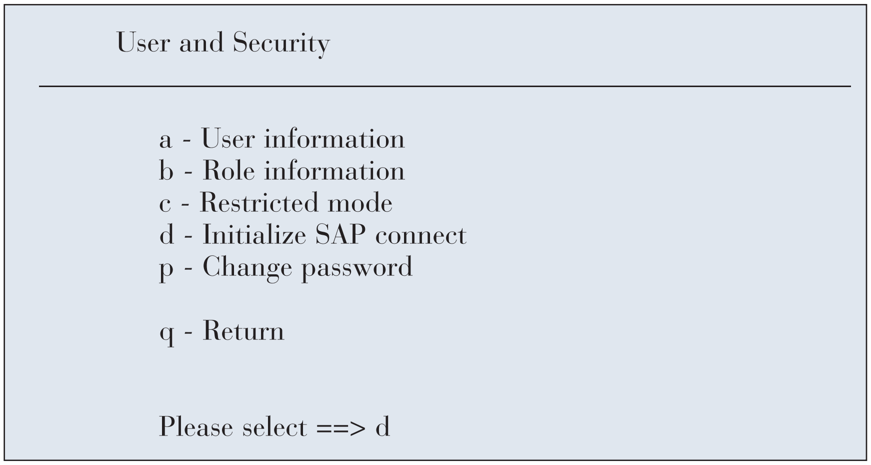
2. Change the instance parameter file for TGT and make parameter rdisp/wp_no_btc equal to 0. This prevents any background jobs that were scheduled from executing. This is a safety precaution so that jobs that could potentially affect the SRC system don't execute accidentally. If SRC was in production, an inadvertent execution of a job could cause a major problem.

3. Start SAP system TGT. As UNIX user tgtadm on host tgthost execute the start SAP scripts (see Figure 13).

4. Install the SAP license for SAP system TGT. See Figure 14. See Appendix for licenseTGT.txt format.

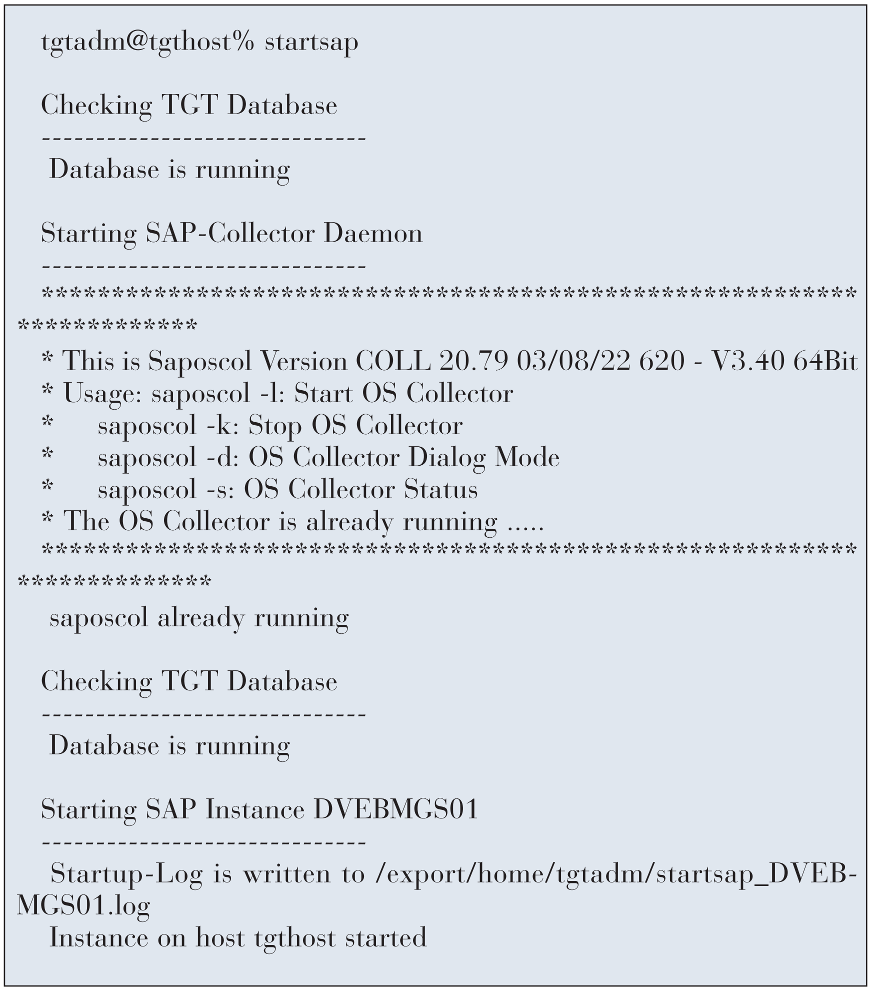
5. Logon to TGT using the SAPGUI and execute transaction code SM28 for an installation check. It is imperative to make sure that on the "Installation Check" screen, no errors are reported (see Figure 15).

6. In order to prevent background jobs from executing,



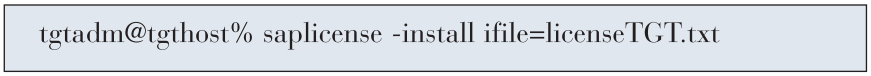
```
          User and Security
   ____________________________________________________

                a - User information
                b - Role information
                c - Restricted mode
                d - Initialize SAP connect
                p - Change password

                q - Return


                Please select ==> d
```

Figure 12a: sapdba Tool, Selecting "Initialize SAP Connect"



```
  tgtadm@tgthost% startsap

  Checking TGT Database
  ------------------------------
   Database is running

  Starting SAP-Collector Daemon
  ------------------------------
  *************************************************************
*************
  * This is Saposcol Version COLL 20.79 03/08/22 620 - V3.40 64Bit
  * Usage: saposcol -l: Start OS Collector
  *     saposcol -k: Stop OS Collector
  *     saposcol -d: OS Collector Dialog Mode
  *     saposcol -s: OS Collector Status
  * The OS Collector is already running .....
  *************************************************************
**************
   saposcol already running

  Checking TGT Database
  ------------------------------
   Database is running

  Starting SAP Instance DVEBMGS01
  ------------------------------
   Startup-Log is written to /export/home/tgtadm/startsap_DVEB-
MGS01.log
   Instance on host tgthost started
```

Figure 13: Start SAP System TGT



```
  tgtadm@tgthost% saplicense -install ifile=licenseTGT.txt
```

Figure 14: Install License for SAP System

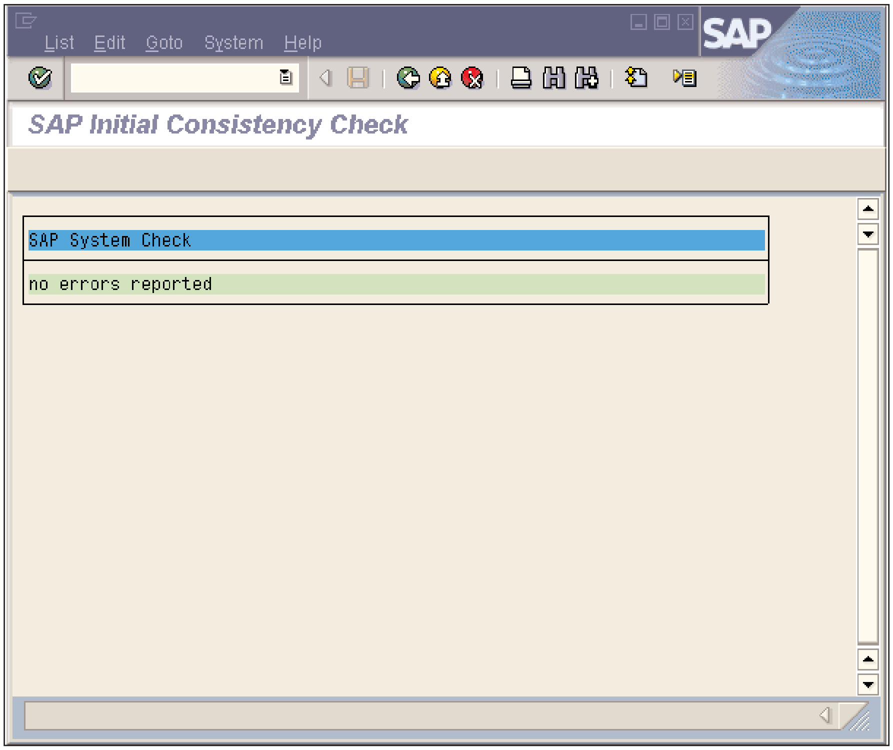
we zeroed the process count of BTC processes in step 2. In this step, we will delete all unnecessary scheduled jobs. The list of required jobs, excluding Basis jobs, should be coordinated with the functional teams. Refer to preparatory step 5 to identify necessary background jobs. Execute transaction code SM37 as shown in Figure 16 and delete all scheduled, released, and ready jobs based on data collected from functional teams.

7. Execute transaction code SM59 and adjust RFC destinations depending on connected SAP and third-party software applications.

8. Adjust the instance profile and increase the parameter rdisp/wp_no_btc to a non-zero number, then restart SAP.

9. Execute SE06 and confirm all pop up messages. You will be initializing the transport mechanism in TGT as well as settings and ownership of objects from SRC to TGT (see Figure 17). Next, import STMS configuration from the TMS DOMAIN controller using transaction STMS. You will need to be in client 000.

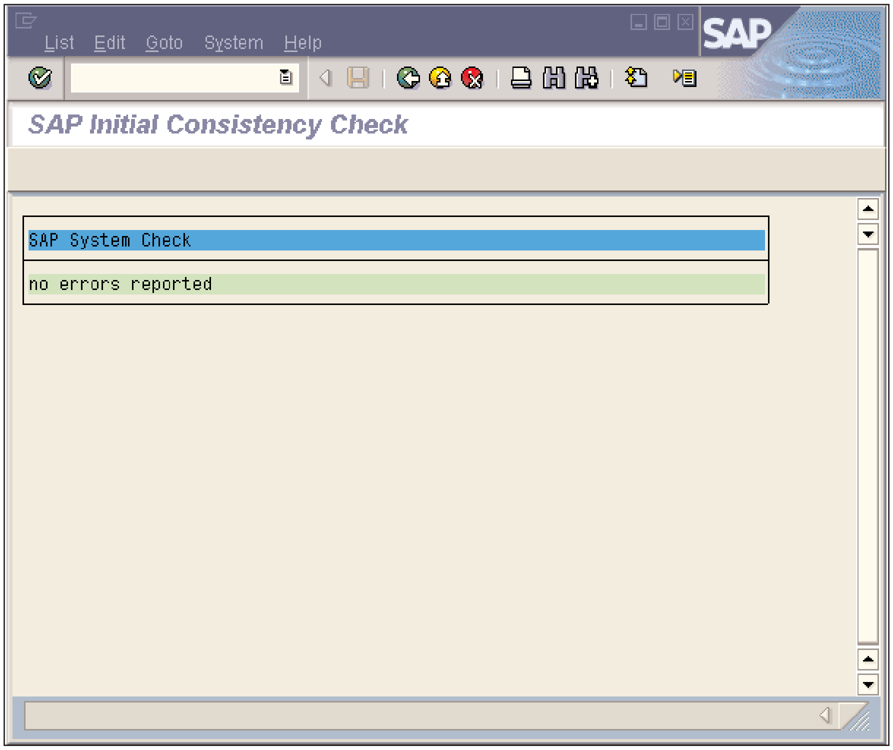
10. Update E070L with the entries recorded in the preparatory phase. This step ensures that TGT keeps tracking with the last transport created. If this step is skipped, the transport counter is reset, which could overwrite any previously released transport prior to the copy.



SAP Initial Consistency Check

```
SAP System Check
no errors reported
```

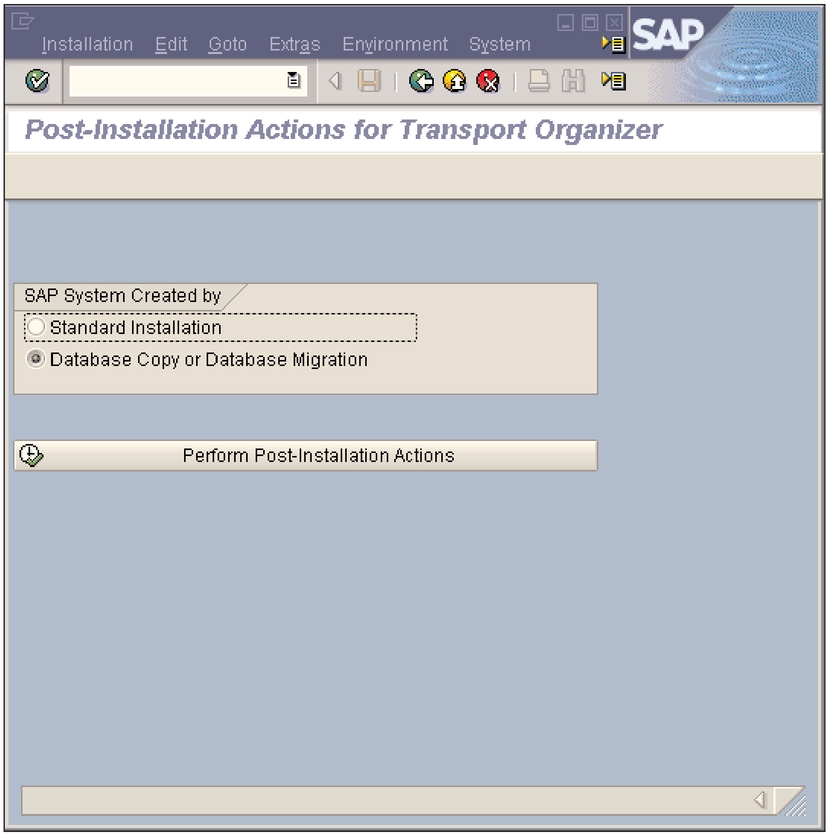
Figure 15: Installation Check Screen Showing "No Errors Reported"



SAP Initial Consistency Check

```
SAP System Check
no errors reported
```

Figure 16: Background Job Selection for Deletion



Post-Installation Actions for Transport Organizer

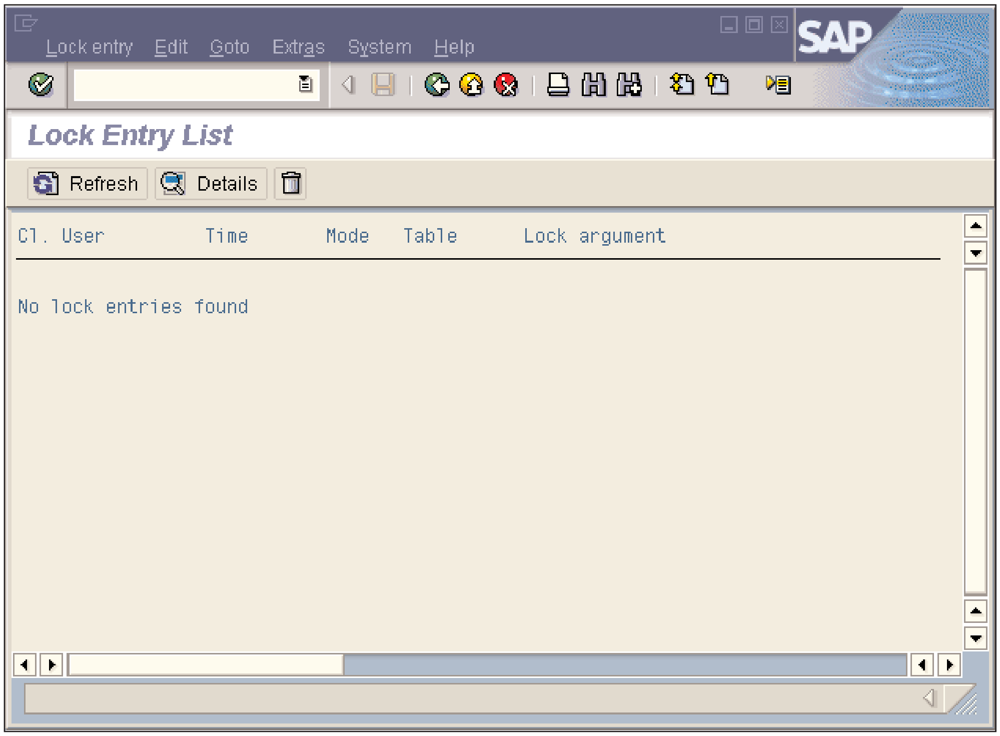
SAP System Created by

○ Standard Installation

● Database Copy or Database Migration

Perform Post-Installation Actions

Figure 17: Executing Transaction SE06



Lock Entry List

```
Cl. User          Time        Mode    Table       Lock argument

No lock entries found
```

Figure 18: Lock Entries List Showing "No Lock Entries Found"

SAPtips Journal

11. Adjust Logon Banner page via SE61 if the logon screen was customized.

12. Delete all locks using transaction code SM12. If no locks are present, continue with the next step. See Figure 18 as an example of "no entries found."

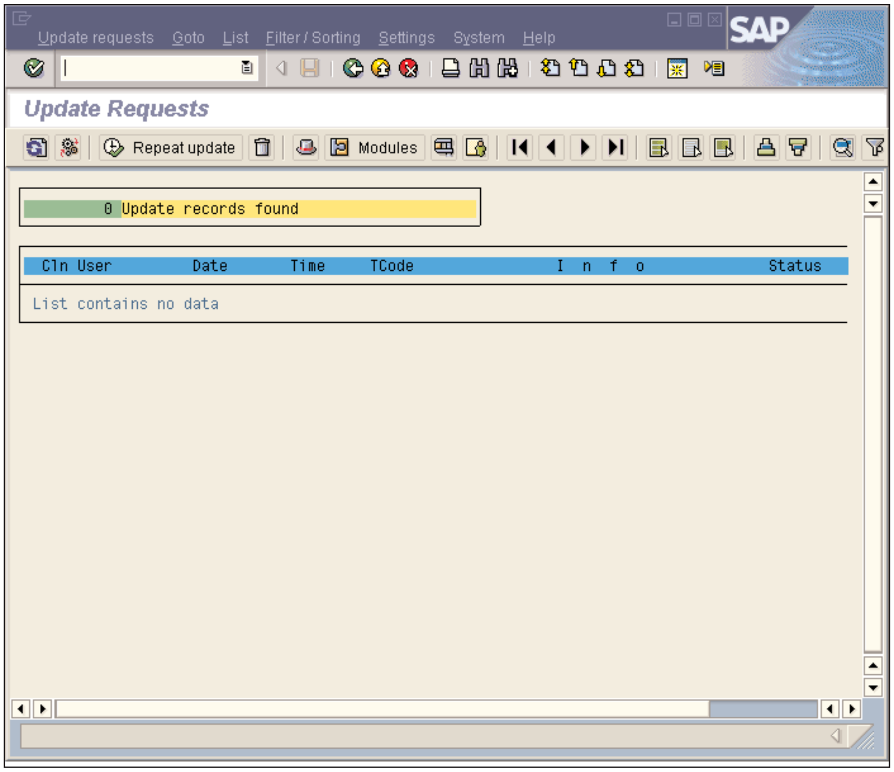
13. Delete all canceled and pending update requests via transaction SM13 (see Figure 19).

14. Delete all transactional RFC entries using SM58 (see Figure 20).

15. Change the logical system name using BDLS (see Figure 21). Specific transactional data is tied to a logical system name that is necessary for specific SAP business functionality. Depending on the size and girth of your system, this process may need a considerable amount of time. Consequently, allocate adequate space for rollback/UNDO segments and time on the project plan. I specifically mention time, since a recent customer had a runtime of over six hours with BDLS. Consequently, ensure you execute BDLS in the background.

16. After BDLS finishes, change the description of the logical system name using SM31 and view name v_tbdls..

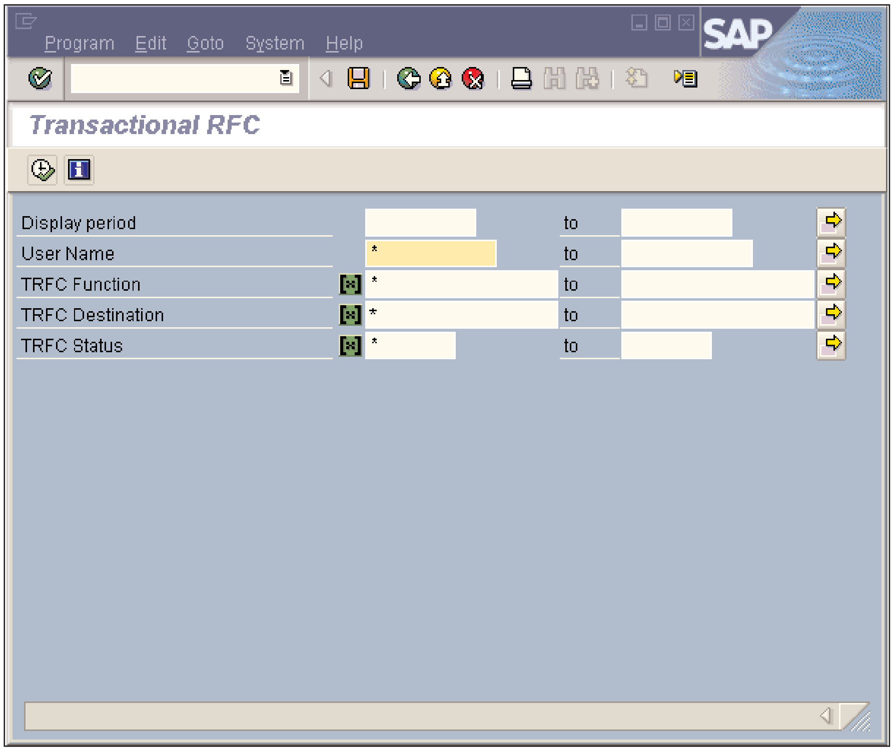
17. Adjust print server for output devices using transaction code SPAD. Lock or delete all printers that you do not want exposed in TGT.



Update Requests

```
       0 Update records found

 Cln User         Date         Time      TCode                  I  n  f  o               Status

 List contains no data
```

Figure 19: Update Requests Screen



Transactional RFC

Display period		to	
User Name	*	to	
TRFC Function	*	to	
TRFC Destination	*	to	
TRFC Status	*	to	

Figure 20: Transactional RFC Screen

18. Import SAP profiles using RZ10.

19. Adapt Logon groups to TGT system requirements using SMLG.

20. Adapt Operations modes to TGT system requirements using SM63.

21. If required, import users using transport identified in preparatory steps using STMS. Execute post-import steps via SCC7. Subsequently, execute PFUD to reconcile user master records.

22. Reschedule administrative jobs via SM36.

23. Schedule backups and update statistic jobs for the database table.

24. Perform any business-required post procedures. For example, secure HR to prevent unauthorized access to live HR data.

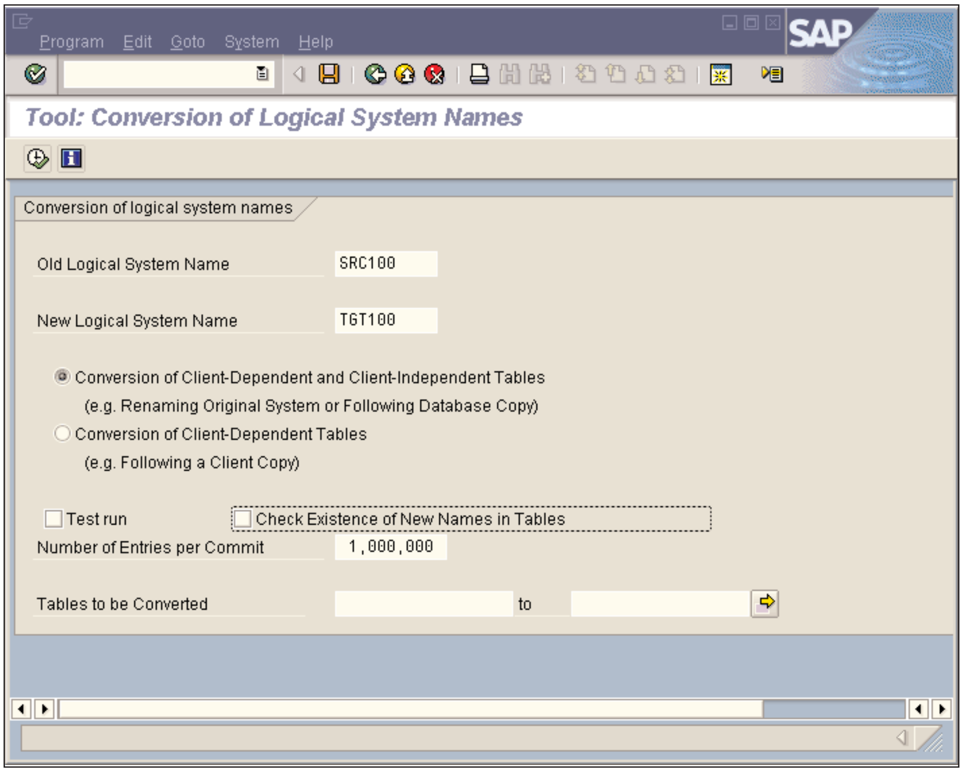
25. Set system change options in the available clients via transaction code SCC4.

26. As a technique, before releasing the newly created system for general use, have a colleague review your work to ensure no steps were overlooked. Once you've got the OK from your colleagues, your re-creation job is complete.

Conclusion

The database copy is a very useful business tool. The technical complexity is medium, but the business benefit is high. A frequent and periodic copy back procedure for the test system from production is a big plus. There is nothing better than testing on an exact copy of your existing production system.

Joey Hirao, *Jotech LLC*. Joey is a Basis consultant with expertise in SAP Basis, UNIX, NT, and database technologies. He designs, implements and maintains SAP systems for customers worldwide. Joey has been working with SAP technology for the past six years. He is the founder of Jotech LLC, and the author of SAP R/3 Administration for Dummies. Joey is SAP Basis certified, Solaris Administrator certified, and a MCSE. Joey's email address is **Joey.Hirao@SAPtips.com.**



Tool: Conversion of Logical System Names

Conversion of logical system names

Old Logical System Name SRC100

New Logical System Name TGT100

● Conversion of Client-Dependent and Client-Independent Tables
(e.g. Renaming Original System or Following Database Copy)

○ Conversion of Client-Dependent Tables
(e.g. Following a Client Copy)

□ Test run □ Check Existence of New Names in Tables

Number of Entries per Commit 1,000,000

Tables to be Converted to

Figure 21: BDLS Logical System Name Rename

Appendix: Scripts and Files for Database Copy:

```
src_ora_4474.trc
/oracle/SRC/saptrace/usertrace/sb1_ora_4474.trc
Oracle9i Enterprise Edition Release 9.2.0.4.0 - 64bit Production
With the Partitioning option
JServer Release 9.2.0.4.0 - Production
ORACLE_HOME = /oracle/SRC/920_64
System name: SunOS
Node name: sapisp
Release: 5.9
Version: Generic_112233-11
Machine: sun4u
Instance name: SRC
Redo thread mounted by this instance: 1
Oracle process number: 35
UNIX process pid: 4474, image: oracle@sapisp (TNS V1-V3)

*** SESSION ID:(36.3097) 2004-04-20 11:31:04.850
*** 2004-04-20 11:31:04.850
# The following are current System-scope REDO Log Archival related
# parameters and can be included in the database initialization file.
#
# LOG_ARCHIVE_DEST=''
# LOG_ARCHIVE_DUPLEX_DEST=''
#
# LOG_ARCHIVE_FORMAT=%t_%s.dbf
# REMOTE_ARCHIVE_ENABLE=TRUE
# LOG_ARCHIVE_START=TRUE
# LOG_ARCHIVE_MAX_PROCESSES=2
# STANDBY_FILE_MANAGEMENT=MANUAL
# STANDBY_ARCHIVE_DEST=?/dbs/arch
# FAL_CLIENT=''
# FAL_SERVER=''
#
# LOG_ARCHIVE_DEST_1='LOCATION=/oracle/SRC/oraarch/SRCarch'
# LOG_ARCHIVE_DEST_1='MANDATORY NOREOPEN NODELAY'
# LOG_ARCHIVE_DEST_1='ARCH NOAFFIRM SYNC'
# LOG_ARCHIVE_DEST_1='NOREGISTER NOALTERNATE NODEPENDENCY'
# LOG_ARCHIVE_DEST_1='NOMAX_FAILURE NOQUOTA_SIZE NOQUOTA_USED'
# LOG_ARCHIVE_DEST_STATE_1=ENABLE
#
# Below are two sets of SQL statements, each of which creates a new
# control file and uses it to open the database. The first set opens
# the database with the NORESETLOGS option and should be used only if
# the current versions of all online logs are available. The second
# set opens the database with the RESETLOGS option and should be used
# if online logs are unavailable.
# The appropriate set of statements can be copied from the trace into
# a script file, edited as necessary, and executed when there is a
# need to re-create the control file.
```

```
#
#   Set #1. NORESETLOGS case
#
# The following commands will create a new control file and use it
# to open the database.
# Data used by the recovery manager will be lost. Additional logs may
# be required for media recovery of offline data files. Use this
# only if the current version of all online logs are available.
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "SRC" NORESETLOGS ARCHIVELOG
-- SET STANDBY TO MAXIMIZE PERFORMANCE
  MAXLOGFILES 255
  MAXLOGMEMBERS 3
  MAXDATAFILES 254
  MAXINSTANCES 50
  MAXLOGHISTORY 1361
LOGFILE
 GROUP 1 (
  '/oracle/SRC/origlogA/log_g11m1.dbf',
  '/oracle/SRC/mirrlogA/log_g11m2.dbf'
 ) SIZE 50M,
 GROUP 2 (
  '/oracle/SRC/origlogB/log_g12m1.dbf',
  '/oracle/SRC/mirrlogB/log_g12m2.dbf'
 ) SIZE 50M,
 GROUP 3 (
  '/oracle/SRC/origlogA/log_g13m1.dbf',
  '/oracle/SRC/mirrlogA/log_g13m2.dbf'
 ) SIZE 50M,
 GROUP 4 (
  '/oracle/SRC/origlogB/log_g14m1.dbf',
  '/oracle/SRC/mirrlogB/log_g14m2.dbf'
 ) SIZE 50M
-- STANDBY LOGFILE
DATAFILE
 '/oracle/SRC/sapdata1/system_1/system.data1',
 '/oracle/SRC/sapdata3/r3m_1/r3m.data1',
 '/oracle/SRC/sapdata3/r3m_2/r3m.data2',
 '/oracle/SRC/sapdata3/r3m_3/r3m.data3',
 '/oracle/SRC/sapdata3/r3m_4/r3m.data4',
 '/oracle/SRC/sapdata4/r3m_5/r3m.data5',
 '/oracle/SRC/sapdata4/r3m_6/r3m.data6',
 '/oracle/SRC/sapdata4/r3m_7/r3m.data7',
 '/oracle/SRC/sapdata4/r3m_8/r3m.data8',
 '/oracle/SRC/sapdata4/r3m_9/r3m.data9',
 '/oracle/SRC/sapdata1/r3m620_1/r3m620.data1',
 '/oracle/SRC/sapdata1/r3m620_2/r3m620.data2',
 '/oracle/SRC/sapdata1/r3m620_3/r3m620.data3',
 '/oracle/SRC/sapdata1/r3m620_4/r3m620.data4',
 '/oracle/SRC/sapdata2/r3m620_5/r3m620.data5',
 '/oracle/SRC/sapdata2/r3m620_6/r3m620.data6',
 '/oracle/SRC/sapdata2/r3m620_7/r3m620.data7',
 '/oracle/SRC/sapdata2/r3m620_8/r3m620.data8',
 '/oracle/SRC/sapdata2/r3m620_9/r3m620.data9',
 '/oracle/SRC/sapdata3/r3m620_10/r3m620.data10',
 '/oracle/SRC/sapdata4/r3m620_11/r3m620.data11',
 '/oracle/SRC/sapdata1/r3musr_1/r3musr.data1',
 '/oracle/SRC/sapdata2/r3m_10/r3m.data10',
 '/oracle/SRC/sapdata1/undo_1/undo.data1',
 '/oracle/SRC/sapdata2/undo_2/undo.data2',
 '/oracle/SRC/sapdata2/r3m_11/r3m.data11',
 '/oracle/SRC/sapdata2/r3m_12/r3m.data12',
 '/oracle/SRC/sapdata2/r3m_13/r3m.data13'
CHARACTER SET WE8DEC
;
# Recovery is required if any of the datafiles are restored backups,
# or if the last shutdown was not normal or immediate.
RECOVER DATABASE
# All logs need archiving and a log switch is needed.
ALTER SYSTEM ARCHIVE LOG ALL;
# Database can now be opened normally.
ALTER DATABASE OPEN;
# Commands to add tempfiles to temporary tablespaces.
# Online tempfiles have complete space information.
# Other tempfiles may require adjustment.
ALTER TABLESPACE PSAPTEMP ADD TEMPFILE '/oracle/SRC/sapdata3/temp_1/temp.data1'
   SIZE 2000M REUSE AUTOEXTEND OFF;
# End of tempfile additions.
#
#   Set #2. RESETLOGS case
#
# The following commands will create a new control file and use it
# to open the database.
# The contents of online logs will be lost and all backups will
# be invalidated. Use this only if online logs are damaged.
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "SRC" RESETLOGS ARCHIVELOG
-- SET STANDBY TO MAXIMIZE PERFORMANCE
  MAXLOGFILES 255
  MAXLOGMEMBERS 3
  MAXDATAFILES 254
  MAXINSTANCES 50
  MAXLOGHISTORY 1361
LOGFILE
 GROUP 1 (
  '/oracle/SRC/origlogA/log_g11m1.dbf',
  '/oracle/SRC/mirrlogA/log_g11m2.dbf'
 ) SIZE 50M,
 GROUP 2 (
  '/oracle/SRC/origlogB/log_g12m1.dbf',
  '/oracle/SRC/mirrlogB/log_g12m2.dbf'
 ) SIZE 50M,
 GROUP 3 (
  '/oracle/SRC/origlogA/log_g13m1.dbf',
  '/oracle/SRC/mirrlogA/log_g13m2.dbf'
 ) SIZE 50M,
 GROUP 4 (
  '/oracle/SRC/origlogB/log_g14m1.dbf',
  '/oracle/SRC/mirrlogB/log_g14m2.dbf'
 ) SIZE 50M
-- STANDBY LOGFILE
DATAFILE
 '/oracle/SRC/sapdata1/system_1/system.data1',
 '/oracle/SRC/sapdata3/r3m_1/r3m.data1',
 '/oracle/SRC/sapdata3/r3m_2/r3m.data2',
 '/oracle/SRC/sapdata3/r3m_3/r3m.data3',
 '/oracle/SRC/sapdata3/r3m_4/r3m.data4',
 '/oracle/SRC/sapdata4/r3m_5/r3m.data5',
 '/oracle/SRC/sapdata4/r3m_6/r3m.data6',
 '/oracle/SRC/sapdata4/r3m_7/r3m.data7',
 '/oracle/SRC/sapdata4/r3m_8/r3m.data8',
 '/oracle/SRC/sapdata4/r3m_9/r3m.data9',
 '/oracle/SRC/sapdata1/r3m620_1/r3m620.data1',
 '/oracle/SRC/sapdata1/r3m620_2/r3m620.data2',
 '/oracle/SRC/sapdata1/r3m620_3/r3m620.data3',
 '/oracle/SRC/sapdata1/r3m620_4/r3m620.data4',
 '/oracle/SRC/sapdata2/r3m620_5/r3m620.data5',
 '/oracle/SRC/sapdata2/r3m620_6/r3m620.data6',
 '/oracle/SRC/sapdata2/r3m620_7/r3m620.data7',
 '/oracle/SRC/sapdata2/r3m620_8/r3m620.data8',
 '/oracle/SRC/sapdata2/r3m620_9/r3m620.data9',
 '/oracle/SRC/sapdata3/r3m620_10/r3m620.data10',
 '/oracle/SRC/sapdata4/r3m620_11/r3m620.data11',
 '/oracle/SRC/sapdata1/r3musr_1/r3musr.data1',
 '/oracle/SRC/sapdata2/r3m_10/r3m.data10',
 '/oracle/SRC/sapdata1/undo_1/undo.data1',
 '/oracle/SRC/sapdata2/undo_2/undo.data2',
 '/oracle/SRC/sapdata2/r3m_11/r3m.data11',
 '/oracle/SRC/sapdata2/r3m_12/r3m.data12',
 '/oracle/SRC/sapdata2/r3m_13/r3m.data13'
CHARACTER SET WE8DEC
;
# Recovery is required if any of the datafiles are restored backups,
# or if the last shutdown was not normal or immediate.
RECOVER DATABASE USING BACKUP CONTROLFILE
# Database can now be opened zeroing the online logs.
ALTER DATABASE OPEN RESETLOGS;
# Commands to add tempfiles to temporary tablespaces.
# Online tempfiles have complete space information.
# Other tempfiles may require adjustment.
ALTER TABLESPACE PSAPTEMP ADD TEMPFILE '/oracle/SRC/sapdata3/temp_1/temp.data1'
   SIZE 2000M REUSE AUTOEXTEND OFF;
# End of tempfile additions.
```

```
#

control.sql

STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "TGT" RESETLOGS ARCHIVELOG
  MAXLOGFILES 255
  MAXLOGMEMBERS 3
  MAXDATAFILES 254
  MAXINSTANCES 50
  MAXLOGHISTORY 1361
LOGFILE
 GROUP 1 (
  '/oracle/TGT/origlogA/log_g11m1.dbf',
  '/oracle/TGT/mirrlogA/log_g11m2.dbf'
 ) SIZE 50M,
 GROUP 2 (
  '/oracle/TGT/origlogB/log_g12m1.dbf',
  '/oracle/TGT/mirrlogB/log_g12m2.dbf'
 ) SIZE 50M,
 GROUP 3 (
  '/oracle/TGT/origlogA/log_g13m1.dbf',
  '/oracle/TGT/mirrlogA/log_g13m2.dbf'
 ) SIZE 50M,
 GROUP 4 (
  '/oracle/TGT/origlogB/log_g14m1.dbf',
  '/oracle/TGT/mirrlogB/log_g14m2.dbf'
 ) SIZE 50M
DATAFILE
 '/oracle/TGT/sapdata1/system_1/system.data1',
 '/oracle/TGT/sapdata3/r3m_1/r3m.data1',
 '/oracle/TGT/sapdata3/r3m_2/r3m.data2',
 '/oracle/TGT/sapdata3/r3m_3/r3m.data3',
 '/oracle/TGT/sapdata3/r3m_4/r3m.data4',
 '/oracle/TGT/sapdata4/r3m_5/r3m.data5',
 '/oracle/TGT/sapdata4/r3m_6/r3m.data6',
 '/oracle/TGT/sapdata4/r3m_7/r3m.data7',
 '/oracle/TGT/sapdata4/r3m_8/r3m.data8',
 '/oracle/TGT/sapdata4/r3m_9/r3m.data9',
 '/oracle/TGT/sapdata1/r3m620_1/r3m620.data1',
 '/oracle/TGT/sapdata1/r3m620_2/r3m620.data2',
 '/oracle/TGT/sapdata1/r3m620_3/r3m620.data3',
 '/oracle/TGT/sapdata1/r3m620_4/r3m620.data4',
 '/oracle/TGT/sapdata2/r3m620_5/r3m620.data5',
 '/oracle/TGT/sapdata2/r3m620_6/r3m620.data6',
 '/oracle/TGT/sapdata2/r3m620_7/r3m620.data7',
 '/oracle/TGT/sapdata2/r3m620_8/r3m620.data8',
 '/oracle/TGT/sapdata2/r3m620_9/r3m620.data9',
 '/oracle/TGT/sapdata3/r3m620_10/r3m620.data10',
 '/oracle/TGT/sapdata4/r3m620_11/r3m620.data11',
 '/oracle/TGT/sapdata1/r3musr_1/r3musr.data1',
 '/oracle/TGT/sapdata2/r3m_10/r3m.data10',
 '/oracle/TGT/sapdata1/undo_1/undo.data1',
 '/oracle/TGT/sapdata2/undo_2/undo.data2',
 '/oracle/TGT/sapdata2/r3m_11/r3m.data11',
 '/oracle/TGT/sapdata2/r3m_12/r3m.data12',
 '/oracle/TGT/sapdata2/r3m_13/r3m.data13'
CHARACTER SET WE8DEC
;
alter database open resetlogs;

ALTER TABLESPACE PSAPTEMP ADD TEMPFILE '/oracle/TGT/sapdata3/temp_1/temp.data1'
SIZE 2000M REUSE AUTOEXTEND OFF;

sapuser.sql

CREATE USER "OPS$TGTADM" DEFAULT TABLESPACE PSAPR31USR
TEMPORARY TABLESPACE PSAPTEMP IDENTIFIED EXTERNALLY;

CREATE USER "OPS$ORATGT" DEFAULT TABLESPACE PSAPR31USR
TEMPORARY TABLESPACE PSAPTEMP IDENTIFIED EXTERNALLY;

GRANT CONNECT, RESOURCE TO "OPS$TGTADM";
GRANT CONNECT, RESOURCE TO "OPS$ORATGT";

create table "OPS$TGTADM".SAPUSER as select * from "OPS$SRCADM".SAPUSER;

drop user OPS$SRCADM cascade;
drop user OPS$ORASRC cascade;

licenseTGT.txt

################################################
# Import License Key
# Script for Release 4.5A and Later
# saplicense -install [ifile=...]
################################################
# Database: ORACLE
# Release: 620
################################################
#
SAPSYSTEM = TGT
HARDWARE-KEY = Z0123456789
INSTNO = 123456
EXPIRATION = 99991231
LKEY = HOUWHKOGHNV4LVSV3WGPK4T8
SYSTEM-NR = 000000000011223344

Trunc.sql

truncate table sapr3m.apqd;
truncate table sapr3m.ddlog;
truncate table sapr3m.sdbad;
truncate table sapr3m.sdbah;
truncate table sapr3m.sdbar;
truncate table sapr3m.dbstathora;
truncate table sapr3m.dbstaihora;
truncate table sapr3m.dbstatiora;
truncate table sapr3m.dbstattora;
truncate table sapr3m.dbsnp;
truncate table sapr3m.osmon;
truncate table sapr3m.pahi;
truncate table sapr3m.moni;
truncate table sapr3m.tpfet;
truncate table sapr3m.tpfht;
truncate table sapr3m.tlock;
truncate table sapr3m.sdbap;
truncate table sapr3m.mlicheck;
truncate table sapr3m.alconseg ;
truncate table sapr3m.alsystems ;
```