A Homogeneous System Copy in 60 Minutes? It Can Be Done!

Bert Vanstechelman

Bert Vanstechelman is an independent SAP Basis Consultant and has approximately six years of professional computing experience. His most recent assignments have been in SAP Basis consulting roles. He specializes in SAP hardware infrastructures, high availability, installations, upgrades, backup/recovery issues, and last, but not least, OS/DB migrations.

(complete bio appears on page 28)

Setting up a system to test Euro migration ... A dedicated reporting system ... The setup of a standby database ... Systems used for customizing ... End-user training systems ... Development systems ... Quality assurance systems ... Upgrade and system consolidation projects. What do all of these have in common? They may require you to make a system copy. But making a system copy (e.g., copying an R/3 production system's database to another R/3 host) is a time-consuming endeavor. SAP documentation suggests that the process takes two days. This article describes how to do an automatic, homogeneous system copy[1] in less than 60 minutes.

This is a solution that I devised to automatically copy and rename a production database to a new database. A copy of a production database is a very handy thing to have around. It can be used for a variety of purposes:

- **Provide end users with an offline reporting system.** Many customers create lots of custom reports, which can place quite a load on a production system. If you don't have a means for offloading your production system like SAP Business Information Warehouse (SAP BW), or you simply can't abandon those custom-made R/3 reports just yet, moving them offline can increase response time on your online transaction processing (OLTP) system.
- **Provide customization consultants and developers with an up-to-date copy of the production database for testing and**

[1] SAP recommends that you make a system copy in order to build a test, demo, or training system. A system copy is *homogeneous* if both source and target hosts (the target hosts the database copy) run the same operating system and relational database management system (RDBMS).

training purposes. Consultants and ABAP developers habitually bemoan the lack of viable, comprehensive test data on the quality assurance system (QAS). The procedure that I'll show you offers a fully automated mechanism for refreshing your quality assurance system! And this same procedure can easily be adapted to refresh other "special purpose" systems, e.g., a system used for Euro conversion activities[2] or for training purposes.

- **Provide Informix shops with a secondary database to run the Informix database consistency check.** This database check verifies all data pages on disk. A lot of customers run it on a weekly basis, and this can be burdensome, since the database consistency check requires the database to be offline. Needless to say, performing regular consistency checks is a nearly impossible undertaking for 24x7 shops.

In this article, I will show you first how to manually create a system copy by installing a UNIX and RDBMS environment (these are not the standard SAP steps for installing a new system, so there is a big time savings even before you automate the process) and connecting source and target systems to a disk array that will facilitate, via disk mirroring, the system copy. Then I will show you how to create several scripts that run on the target system to execute an automatic procedure for refreshing the system copy with your production system. These scripts are called by a single script that can be invoked from the command line or scheduled to run on a regular basis.

This solution will work for R/3 systems that run on Informix and Oracle databases. But don't stop reading if your R/3 landscape doesn't operate on either of these platforms. You should be able to easily adapt the measures I will be presenting here for other database platforms.[3]

Plan of Action

Before you can make an automatic system copy, you need an environment to support it — an environment that matches the one on your production system's host (i.e., the same UNIX system, database system, and SAP environment). As part of this one-time setup, you will also make your first copy of the production database and verify that the system copy is working properly. Only then are you ready to make automatic system copies using a scripted procedure. The scripted system copy procedure will copy, in addition to the database itself, the UNIX and RDBMS profiles and configurations, so that the new system copy is always an exact duplicate of the source database and its environment.

Before plunging into the details of how to accomplish all this, let's briefly review the technology we will be using.

The Technology

Most storage vendors (including EMC², HP, and Compaq) provide storage solutions that make it possible to copy large amounts of data quickly, with little (if any) impact on the production system. All of these solutions utilize Business Continuance Volumes (BCVs). A BCV is a disk like any other, which has been configured by the storage vendor for this special purpose. An active production volume (e.g., a database) can be copied to a BCV. The BCV is assigned to a host system that views it as it would any other active volume. Most database vendors, including SAP, support the use of BCV. SAP's split mirror

[2] One of my customers is now using this automatic, homogeneous system copy procedure to support the company's Euro conversion efforts. A central system has been set up, onto which copies of production databases are being made. Since this customer has three independent SAP landscapes, testing is done on a rotating basis. One day, the Euro conversion is tested for one production database, the next day for another.

[3] The Backup/Recovery document for mySAP.com (part of Best Practices for Solution Management, available from SAP at **http://service.sap.com/solutionmanagerbp**) discusses this solution for DB2, Informix, Oracle, and SQL Server.

Why This Setup and This Solution at This Time?

The customer for whom I originally devised the automated system copy solution that you'll be reading about had two pressing needs. The first was for a reporting system that would be refreshed with production data every day. They had just upgraded from R/3 Release 3.1I to 4.6B and were experiencing performance problems, because a 4.6 system needs far more resources than a 3.1I system, and they had grossly underestimated those requirements. In particular, they had not adequately accounted for their extensive collection of custom-made reports, which, as we all know, can place a heavy processing load on the database. Moving those custom reports off the OLTP system would accelerate ailing response times.*

The customer's second pressing need stemmed from the fact that they had implemented the Warehouse Management (WM) module and needed their Informix-based system to be online 24x7. They wanted an alternative to the Informix requirement that the system must be taken offline in order to run database consistency checks.

Copying and renaming the production database onto a dedicated system that could be used expressly for reporting purposes was clearly the answer. Creating scripts to automate the refresh of the system copy database with the current production database would make maintaining the copy both efficient and convenient.

Note that while this solution was devised for an SAP production database on an HP9000 N-Class, HP-UX 11 (64-bit) system, with a 400 GB Informix database installed on EMC² 8470 disks in a Storage Area Network (SAN), the procedure can easily be adapted for other UNIX versions and storage solutions with or without a SAN.

* If you're wondering whether or not this client had plans for BW, the answer is yes. A BW implementation takes time, however, and they needed those custom reports to be up and running then, not later. Today, the customer is implementing BW.

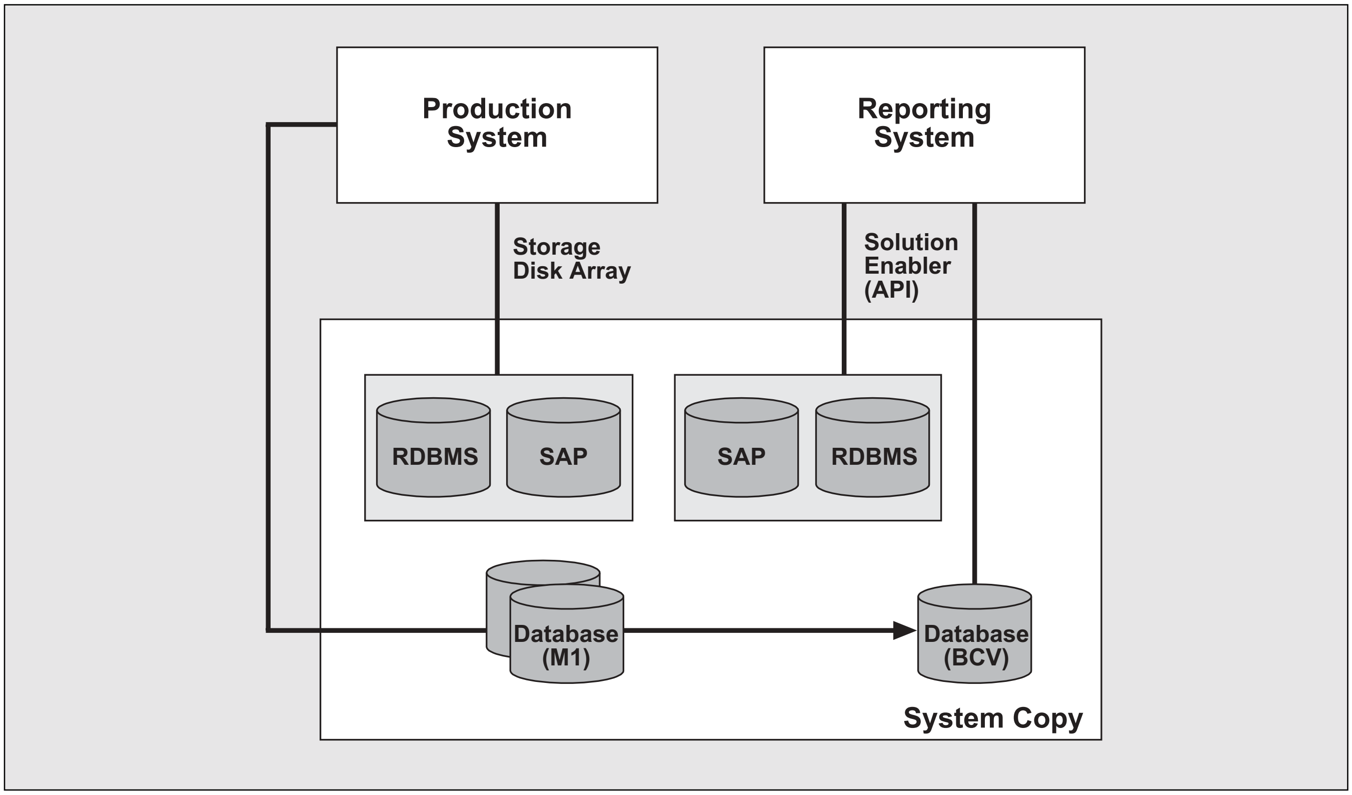
disk backup solution for Oracle, for example, has supported BCV via the BRBACKUP function since Release 4.0.[4]

To copy a production volume to a BCV, the BCV can be temporarily connected to a primary disk (M1). An M1 is a mirrored disk configured by the storage vendor such that there are two physical disks, but the system sees only one disk (M1). When the BCV is temporarily connected to M1, it becomes a third mirror on the source host. During this time, it is not visible to its host system (the target system). Before data is copied to the BCV, the mirrors are "split," which means that one of the mirrored disks continues to function online while the other is being used as the source for a point-in-time copy. After the copy is completed, the mirrored disks are "joined" (re-synchronized). The BCV becomes visible to the target host again after disconnecting from the primary disk.

BCV and M1 disks can be grouped together in device groups, which makes it easier to manipulate them. The operating system manipulates the disks using an API that is provided by the storage vendor. This API enhances UNIX with tools to create and manipulate devices. EMC² calls its API the "Solution Enabler." HP calls it the "Business Copy Solution." Regardless of the appellation, it's effectively the same technology.

[4] There is a very interesting white paper from SAP on the split mirror backup solution for Oracle at **http://service.sap.com/dbainf**.

Figure 1 *The Environment*



Both systems — the production system and the system onto which you will be placing a copy of the production system's database — are connected to the same disk array. You can see this in **Figure 1**, where we are using a system copy to create an offline reporting system. Each host has its own disks for the Relational Database Management System (RDBMS), SAP executables, and log files. The storage disk array is used to mirror the disks between the source and target database disks.

Before splitting the mirrors, you need to ensure that the database copy on the target host is consistent. In other words, you don't want the mirrored disk to reflect a production database that is in the middle of performing transactions and updates, because it would take too long for the database copy to do an online recovery (undo the uncommitted transactions) before the database is available. Or worse, if the database has too many uncommitted transactions, it might not come up at all.

An Informix instance can be "blocked," which puts it in read-only mode. Before switching to read-only mode, Informix will flush all updates to disk. While the database is in read-only mode, all updates are buffered in memory. There is no impact for the application (SAP) if the read-only mode takes no longer than a few minutes. During the split, the read-only mode takes no longer than 60 seconds. The solution, then, is to block the Informix instance on the production host before splitting the mirrors.

Unfortunately, Oracle is not that flexible. Before splitting the volumes, you should put the production database in backup mode using a small script that connects to the database via Oracle's Server Manager and switches its tablespaces (self-contained sets of disk files and stored data) into backup mode.[5] Unlike

[5] There seems to be a new command for Oracle (`alter system suspend/resume`) that provides the same functionality as the `block/unblock` mode for Informix. There was no time to test it, however, before sending this article to press.

the Informix database copy, the Oracle database copy will be inconsistent after the split, but the backup mode will force Oracle to do an automatic recovery on the reporting host as soon as it comes up. (After the split, you connect to the production database with the same script to perform an end backup.) The SAP BRBACKUP function does more or less the same in SAP's mirror disk backup solution for Oracle.

Setting Up the Source and Target R/3 Systems and Making Your First System Copy

Time to get started. Follow these 10 steps to set up your source and target systems and make your first system copy:

1. Install a UNIX host. This will be the target system on which the copied database will reside. The newly installed UNIX box and your production system (the source system) must both be running the same operating system. The target system's hardware, however, does not need to be as high-performing as the source system's. (In my case, I did add extra CPUs to the target system to ensure fast response times for reporting.) Now, connect your target system to the disk array and assign to it the same number of disks used by the database on the production system. This is a one-time setup step. Your automatic system copy scripts will not need to perform this step.

2. On the target system, install the API that will communicate with the disk array. In the example, I am using the Solution Enabler from EMC². You will want to consult your hardware vendor's documentation on the API that is appropriate for your particular disk array. This is another one-time-only job.

3. Create a device group called `<SID>_dev` on the target. Assign the primary disk M1 and the disk BCV to this device group. The Solution Enabler commands for creating the device group are shown in **Figure 2**. This too is a one-time setup task.

Figure 2 Solution Enabler Commands to Create the Device Group

✓ **Create a new device group:**

```
# /usr/symcli/bin/symdg -type regular create <SID>_dev
example # /usr/symcli/bin/symdg -type regular create RPT_dev
```

✓ **Add standard devices:**

```
# /usr/symcli/bin/symld -g <SID>_dev add dev <EMC² dev>
example # /usr/symcli/bin/symld -g RPT_dev add dev 087
```

✓ **Associate BCV with standard devices:**

```
# /usr/symcli/bin/symbcv -g <SID>_dev associate pd <pdev>
example # /usr/symcli/bin/symbcv -g RPT_dev associate pd c0t2d0
```

✓ **Check device group:**

```
#/usr/symcli/bin/symdg show <SID>_dev
example # symdg show RPT_dev
```

(continued on next page)

Figure 2 (continued)

✓ **Fully establish the device group for the first time:**

```
# /usr/symcli/bin/symmir -g <SID>_dev -full establish
example # /usr/symcli/bin/symmir -g RPT_dev -full establish DEV001 BCV ld BCV001
```

✓ **Verify establishing process:**

```
# /usr/symcli/bin/symmir -g <SID>_dev query
example # /usr/symcli/bin/symmir -g RPT_dev query
```

✓ **Split a device group:**

```
# /usr/symcli/bin/symmir -g <SID>_dev query
example # /usr/symcli/bin/symmir -g RPT_dev split
```

✓ **Establish a device group:**

```
# /usr/symcli/bin/symmir -g <SID>_dev establish
example # /usr/symcli/bin/symmir -g RPT_dev establish
```

4. Now that you have an environment for the reporting database, you are ready to copy it to the target system. Mirror the disks, and then split them. Establishing the disks can take some time, but the split should not take longer than a few minutes.[6]

   For the sake of simplicity, just stop SAP and the database on the production host before splitting the mirrors. (This is in lieu of blocking your Informix instance or putting your Oracle database into backup mode. The script that executes this step in the automatic system copy procedure will, however, either block Informix or put Oracle in backup mode.)

5. On the source system, export the UNIX volume group configuration and then import it on the target system (don't forget to change the ownership of the raw devices if you are using Informix). Every UNIX flavor has its own way of handling volume groups. The commands are different but the technology is more or less the same. HP-UNIX uses the commands `vgexport` and `vgimport`. Consult your vendor's documentation for more information on the logical volume manager (LVM) for your OS. This step will be repeated each time you make a system copy.

The remaining steps complete the setup of your target system by restoring the configurables and executables from the production to the target system. They only have to be performed one time.

6. To make the installation easier, restore the Informix or Oracle system and the SAP file system from the production to the target system and mount these systems under the new `<SID>`:

   - For Informix, restore `INFORMIXDIR`.
   - For Oracle, restore everything under `ORACLEHOME` except the SAP data directories. You should be able to mount the SAP data directories. They were copied during the disk copy in step 4. Don't restore the `REDO LOG FILES`. They will be re-created together with the `CONTROL FILES`. Also restore the SAP file systems `/sapmnt/<SID>` and `/usr/sap/<SID>`. Mount `/usr/sap/trans` from the central instance. Don't forget to change the links in `/usr/sap/<SID>/SYS`. They should point to the new `/sapmnt/<SID>`.[7]

[6] It took 2 hours to establish the disks of a 400 GB database. All sequential establishes are incremental and take no longer than 20 minutes.

[7] Just copying Oracle won't work with Oracle versions higher than 8.1, as Oracle keeps an inventory of all installations. If you are running an Oracle version higher than 8.1, do an installation on your target host and afterward copy the DBS, database, and network/admin directories from your production host to your target host.

Figure 3 Configuration Files That Must Be Changed on the Reporting Host

Informix database → adapt and rename the following files in $INFORMIXDIR/<SID>/etc:

- onconfig.<HOSTNAME>.<SID>
- oncfg_source<SID>shm.<SERVERNUMBER>
- sqlhosts

Oracle database → adapt and rename the following files in $ORACLE_HOME/dbs:

- init<SAPSID>.ora
- init<SAPSID>.dba
- init<SAPSID>.sap
- listener.ora ($ORACLE/network/admin)
- tnsnames.ora ($ORACLE/network/admin or specified by environment variable TNS_ADMIN)

SAP instance:

- Start, default, and instance profiles
- Create new start and stop scripts

Also, copy and rename users <SID>adm, informix, or ora<SID>. Don't forget to change their environment variables in files dbenv_hostname.sh and sapenv_hostname.sh.

7. Copy users `Informix` and `sapr3` (Informix), or `ora <SID>` (Oracle) and `<SID>adm` (SAP administrator) from the production to the target host. Rename them if necessary, and perform the following tasks:
   - Adapt their environment files `dbenv_hostname.sh` and `sapenv_hostname.sh`.
   - Change the SID, instance number, and hostname in both files. It is a quick and dirty solution but it ensures that production and target systems use the same SAP kernel and database version.
   - Create soft links between the old `/Informix/<SID>` and new `/Informix/<SID>`. The links suppress the need to rename the chunks.[8] (You can skip this step in an Oracle environment. The control files will be re-created anyway.)
   - Copy the Informix (or Oracle) and SAP entries from the services file from your production system to your new target system.

8. On the target system, adapt the configuration files listed in **Figure 3**. Change the SID, instance number, and hostname in all files.

9. Create new default, instance, and start profiles on the target system. You already restored them from your production host in step 6. Just rename them and change the SID, instance number, and hostname in each file. Also rename and change the start and stop scripts in the home directory of user `<SID>adm`.

If you are using Oracle, you'll need to take an additional step here (see sidebar on next page).

[8] A *chunk* is the unit amount of contiguous data held on each disk, the equivalent of data files in Oracle.

If You Are Using Oracle...

You'll need to take this extra step:

9A. Re-create the control files and rename the database:

- With Oracle Server Manager (svrmgrl), connect to your production database as user ora<sid>.
- Use the command "alter database backup controlfile to trace". It will generate an SQL file in /oracle/<SID>/saptrace/usertrace/ora_<pid>.trc. You can use this file to re-create the control files.
- Copy ora_<pid>.trc to CONTROL.SQL.
- In CONTROL.SQL, remove the lines preceding "startup nomount".
- Change this line:

  ```
  CREATE CONTROLFILE REUSE DATABASE "SID" NORESETLOGS ARCHIVELOG
  ```

  so that it reads:

  ```
  CREATE CONTROLFILE REUSE SET DATABASE "TARGETSID" RESETLOGS ARCHIVELOG
  ```
- Change all occurrences of SID to TARGETSID.
- Change the line:

  ```
  ALTER DATABASE OPEN
  ```

  to:

  ```
  ALTER DATABASE OPEN RESETLOGS
  ```
- Check to see that your control file (CONTROL.SQL) looks like the example code shown in **Listing 1**.
- Copy the file to the reporting (target) system, and execute it from within the Oracle Server Manager. It will re-create the control files and open the database.

You'll find more information on this step in the "R/3 Homogeneous System Copy Manual," which is available from the installation media section at the SAP Service Marketplace (**http://service.sap.com**).

10. Complete the installation. For Informix, use SAP Note 173970. For Oracle, use the Oracle copy method and the post-processing chapter of the "R/3 Homogeneous System Copy Manual." You should end up with a new running SAP instance.

✔ ***Tip***

When you're all done, don't forget to request a permanent SAP license in OSS!

Congratulations! You have your first system copy! The environment of your target system matches the source system environment, and a running copy of your production database is on the target system. Everything is in place for making future system copies. The next time you make a copy you will only need to copy the source database itself and the UNIX and RDBMS configurations. Now you are ready to create some scripts to automate the process.

Because it takes time to set up the automated process, if you only want a one-time system copy,

Listing 1: Example CONTROL.SQL File for Oracle

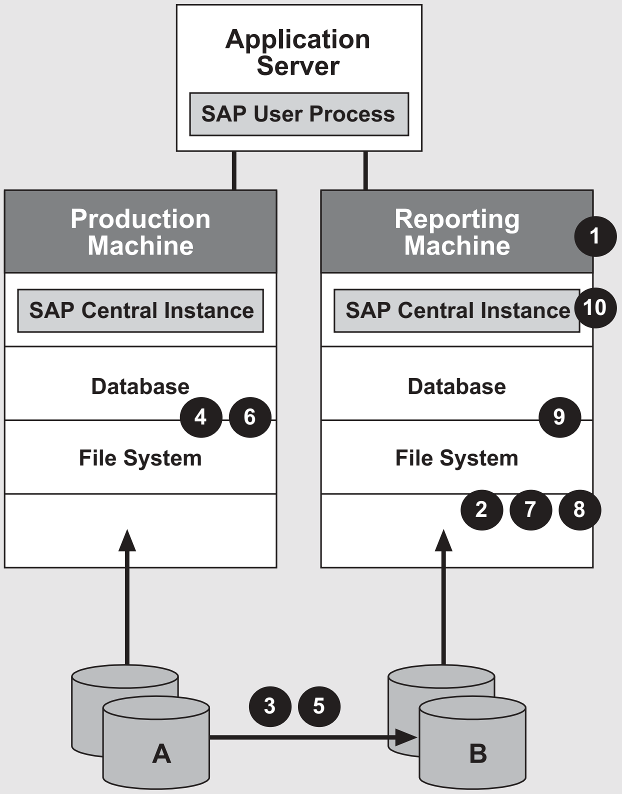
```
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE SET DATABASE "RPT" RESETLOGS ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 1022
    MAXINSTANCES 50
    MAXLOGHISTORY 5218
LOGFILE
  GROUP 11 (
    '/oracle/RPT/mirrlogA/log_g11m2.dbf',
    '/oracle/RPT/origlogA/log_g11m1.dbf'
  ) SIZE 20M,
  GROUP 12 (
    '/oracle/RPT/mirrlogB/log_g12m2.dbf',
    '/oracle/RPT/origlogB/log_g12m1.dbf'
  ) SIZE 20M,
  GROUP 13 (
    '/oracle/RPT/mirrlogA/log_g13m2.dbf',
    '/oracle/RPT/origlogA/log_g13m1.dbf'
  ) SIZE 20M,
  GROUP 14 (
    '/oracle/RPT/mirrlogB/log_g14m2.dbf',
    '/oracle/RPT/origlogB/log_g14m1.dbf'
  ) SIZE 20M
DATAFILE
  '/oracle/RPT/sapdata2/system_1/system.data1',
  '/oracle/RPT/sapdata7/temp_1/temp.data1',
  '/oracle/RPT/sapdata1/roll_1/roll.data1',
  '/oracle/RPT/sapdata2/pooli_1/pooli.data1',
  '/oracle/RPT/sapdata4/ddicd_1/ddicd.data1',

 . . .

  '/oracle/RPT/sapdata12/es46cd_3/es46cd.data3',
  '/oracle/RPT/sapdata1/es46ci_1/es46ci.data1',
  '/oracle/RPT/sapdata1/es46ci_2/es46ci.data2',
  '/oracle/RPT/sapdata12/el46cd_1/el46cd.data1',
  '/oracle/RPT/sapdata1/el46ci_1/el46ci.data1',
  '/oracle/RPT/sapdata3/stabi_6/stabi.data6',
  '/oracle/RPT/sapdata4/ddicd_4/ddicd.data4'
;
ALTER DATABASE OPEN RESETLOGS;
```

or you only plan to make a system copy once or twice a year, you can stop right here. The next time you want to copy your system, just follow steps 4, 5, and 10.

Figure 4 *The Automatic System Copy Procedure*



Unless otherwise noted, the following steps are performed by the script RPT_refresh.sh:

1. Stop SAP on the target host, which in this case functions as a reporting host.
2. Deactivate the volume groups on the reporting host (unmount SAP data if it is an Oracle environment).
3. Establish the mirrors (BCV).
4. Depending on the operating system, block Informix or switch Oracle into backup mode on the production host (this step is performed by the script split-mirror.sh).
5. Split the disks.
6. Unblock Informix. Perform an “end backup” for Oracle (split-mirror.sh again).
7. Refresh the UX volume groups.
8. If this is an Informix environment, refresh chunks (call inf_ux.sh). If it is an Oracle environment, mount SAP data file systems (call ora_ux.sh).
9. Refresh and clean up the database (for Informix, call inf_refresh.sh; for Oracle, call ora_refresh.sql).
10. Start SAP on the target (reporting) host and start the cleanup jobs (this step is performed by sappost.sh).

Creating Scripts to Perform an Automatic System Copy Procedure

The automatic system copy procedure relies on one script (`RPT_refresh.sh`), which calls several other scripts to perform specific tasks. You can invoke `RPT_refresh.sh` from the command line whenever you want to make a new system copy, or you can use your job scheduler to run it at regular intervals. **Figure 4** shows, in sequential order, the most important steps performed by this script and the scripts that it calls. All these scripts run on the target system.

Let's look at these steps in more detail by walking through the `RPT_refresh.sh` script shown in **Listing 2**. (Note that steps 1-8 are performed on the UNIX system, and steps 9 and 10 take place at the RDBMS level.) The code is line-numbered for easy reference.

> ***Warning!***
>
> *You must set up the environment for your system copy as outlined in the previous section ("Setting Up the Source and Target R/3 Systems and Making Your First System Copy") and verify that it works before automating the process by adapting the scripts shown here. Otherwise, you risk investing time in setting up a process that may result in a system copy that doesn't work.*

Let's now walk through the code:

1. The script shuts down SAP on the target (reporting) host (lines 12-13).
2. Next, it deactivates the volume groups on the target host (lines 20-28). In an Oracle environment, it first unmounts all the SAP data

Listing 2: Refresh UNIX and Database (RPT_refresh.sh)

```
 1 #!/usr/bin/sh
 2 #
 3 # RPT_refresh.sh
 4 #
 5 # Author: Bert Vanstechelman
 6 #

 7 exec > /tmp/RPT_refresh.log.$$ 2>&1


 8 VGGROUPS=$(ls -d /dev/vgrptdata* | cut -f3 -d"/")
 9 HOST=$(hostname)

10 date

11 # Stop SAP

12 su - rptadm -c "/home/rptadm/stopsap_zhpux31_05;saposcol -k"
13 sleep 120

14 # for Oracle unmount the SAP data filesystems
15 bdf | grep -e "/oracle/RPT/sapdata" | awk '{print $1}' | while read line
```

(continued on next page)

(continued from previous page)

```
do
  umount $line
done

# disable the UNIX volume groups
for VGGROUP in ${VGGROUPS}
do
  vgchange -a n ${VGGROUP}
  if [ $? -ne 0 ]
  then
     echo "Could not de-activate volume group ${VGGROUP}"
     exit 12
  fi
done

# Establish the Business Continuance Volumes with their primary disks

/usr/symcli/bin/symcfg discover
/usr/symcli/bin/symmir -g RPT_dev establish -noprompt

while [ $(/usr/symcli/bin/symmir -g RPT_dev query | grep -e "SyncInProg" | wc -l) -gt 0 ]
do
  echo "BCV-Syncro still running!"
  sleep 180
done

# for Informix
remsh zhpux5 -l inforprd -n "onmode -c block"

# for Oracle
remsh zhpux5 -l oraprd -n "/usr/local/bin/split-mirror.sh begin"

/usr/symcli/bin/symmir -g RPT_dev split -noprompt
if [ $? -ne 0 ]
then
   echo "Could not split device group"
   # for Informix
   remsh zhpux5 -l inforprd -n "onmode -c unblock"
   # for Oracle
   remsh zhpux5 -l oraprd -n "/usr/local/bin/split-mirror.sh end"
   exit 12
fi

# for Informix
remsh zhpux5 -l inforprd -n "onmode -c unblock"

# for Oracle
remsh zhpux5 -l oraprd -n "/usr/local/bin/split-mirror.sh end"

# refresh the UNIX volume groups
```

```
57 for VGGROUP in ${VGGROUPS}
58 do
59   VGR_SRC=$(echo ${VGGROUP} | sed s/rpt/prd/g )
60   MKNOD=$(ll /dev/${VGGROUP}/group | awk '{print $6}' | cut -f3 -d"0")
61   vgexport ${VGGROUP}
62   mkdir /dev/${VGGROUP}
63   mknod /dev/${VGGROUP}/group c 64 0x0${MKNOD}0000
64   remsh zhpux5 -n "/etc/vgexport -p -m /tmp/${VGR_SRC}.map -s -v
       ${VGR_SCR}"
65   rcp -p zhpux5:/tmp/${VGR_SRC}.map /tmp
66   vgimport -v -m /tmp/${VGR_SRC}.map -s /dev/${VGGROUP}
67   # change permissions for Informix
68   chmod 660 /dev/${VGGROUP}*/rlvdata*
69   chown informix:informix /dev/${VGGROUP}*/rlvdata*
70   vgchange -c n ${VGGROUP}
71   vgchange -a y ${VGGROUP}
72   if [ $? -ne 0 ]
73   then
74       echo "Could not activate volume group ${VGGROUP}"
75       exit 12
76    fi
77    done
78 done

79 # refresh Informix chunks
80 /usr/local/bin/inf_ux.sh

81 # refresh Oracle filesystems
82 /usr/local/bin/ora_ux.sh

83 # post activities

84 # post-processing on RDBMS level (Informix)
85 su - inforprt -c /usr/local/bin/inf_refresh.sh

86 # post-processing on RDBMS level (Oracle)
87 su - oraprt -c "svrmgrl command=@/usr/local/bin/ora_refresh.sql
88 # sap-processing on RDBMS level
89 su - rptadm -c /usr/local/bin/sappost.sh

90 date
```

directories on the target host `/oracle/<SID>/sapdata` (lines 15-18).

3. `RPT_refresh.sh` now establishes the mirrors (the BCVs) in lines 30-36. The script has to use an API to re-synch and split the device groups. In our example it is the Solution Enabler API from EMC^2.

4. Before splitting the device group, the script does one of the following, depending on the environment:

Listing 3: Script to Begin/End Backup Mode for Oracle

```
# Oracle split-mirror.sh script
#
# Author: Bert Vanstechelman
#
# Script to begin or end backup. Takes one
# parameter (begin to begin/end to end).

SCRIPT=/tmp/split-mirror.sql
SPOOL=/tmp/tablespace.lst

rm ${SCRIPT}
rm ${SPOOL}

svrmgrl <<EOF
connect internal
set termout on
set echo off
spool ${SPOOL}
select 'alter tablespace ' ||tablespace_name|| ' $1 backup;' from
  dba_tablespaces;
spool off
!grep "[A-Z]" ${SPOOL} | grep -v "TABLESPACE_NAME" > ${SCRIPT}
@${SCRIPT}
EOF
```

- For Informix, it blocks the Informix instance on the source system (`onmode -c block`, as user `Informix`, on line 38).
- For Oracle, it calls script `split-mirror.sh` on line 40 to connect to the database and switch all tablespaces to backup mode on the source system (`split-mirror.sh` is shown in **Listing 3**). This step ensures that the copy is consistent.

5. The script splits the disks (line 42).

6. Now it unblocks Informix (`onmode -c unblock`, as user `Informix`, on line 53) or it ends backup for Oracle by calling script `split-mirror.sh` again, on line 55).

7. Refresh the UX volume groups on the target system (lines 57-58). The script deactivates and deletes the current volume group configuration and imports the new one from the production host. This is so we can pick up any changes that have been made to the volume group configuration on the production host. If logical volumes were added or deleted on the production host, they should automatically be created or removed on the target system during a system refresh.

8. Now refresh the data in the target system:
   - For Informix, the script calls `inf_ux.sh` on line 80 (the script is shown in **Listing 4**). This script refreshes the links in `/informix/<SID>/physdev/sapdata`. These are the links we created earlier, when we set up the source and target systems and made our original manual system copy (step 7 of the previous section).
   - For Oracle, the script calls `ora_ux.sh` on line 82. This script (shown in **Listing 5**)

Listing 4: Refresh the Informix Links

```
#!/usr/bin/sh
#
# inf_ux.sh
#
# Author: Bert Vanstechelman
#

# Refresh the Informix links
rm –rf /informix/RPT/sapdata/*
remsh zhpux5 -n "ls -d /informix/PRD/sapdata/*" | sed s/PRD/RPT/g | xargs mkdir

remsh zhpux5 -n "ll /informix/PRD/sapdata/phys*/data*" \
   | awk '{print $11,$9}' | sed –e 's/PRD/RPT/g' -e 's/prd/rpt/g' | xargs ln -s

# Refresh Informix oncfg file
rcp -p zhpux5:/informix/PRD/etc/oncfg_zhpux5prdshm.5
/informix/RPT/etc/oncfg_zhpux31rptshm.5
```

Listing 5: Refresh the Oracle CONTROL.SQL File and the Oracle Mount Points

```
#!/usr/bin/sh
#
# ora_ux.sh
#
# Author: Bert Vanstechelman
#

# Refresh mount points
remsh zhpux5 -n 'cat /etc/fstab' | grep -e "/oracle/PRD/sapdata" \
                | sed -e 's/vgprddata/vgrptdata/g' -e 's/PRD/RPT/g' | \
while read LINE do
do
  SAPDATA=$(echo ${LINE} | awk '{print $2}')
  test –d ${SAPDATA}
  if [ $? –ne 0 ]
  then
     mkdir ${SAPDATA}
     echo ${LINE} >> /etc/fstab
  fi
done

mount -a
chmod 775 /oracle/RPT/sapdata*
```

(continued on next page)

(continued from previous page)

```
chown –R orarpt:dba /oracle/RPT/sapdata*

# Create new control file
remsh zhpux5 -n 'su - oracle -c "svrmgrl << EOF
connect internal
archive log current;
alter database backup controlfile to trace;
spool /tmp/SCN.number;
select sequence# from v$log where status='CURRENT';
spool off;
EOF"'

SCN=$(remsh zhpux5 –n "cat /tmp/SCN.number" | grep -v -e '[A-z]' -e '-')

TRACE=$(remsh zhpux5 -n 'ls -tr /oracle/PRD/saptrace/usertrace/*' | tail -n 1)
rcp -p zhpux5:${TRACE} /oracle/RPT/sapreorg/TRACE.SQL

cat /oracle/RPT/sapreorg/TRACEL.SQL | grep -v -e "#" | tail -n +18 \
    sed -e 's/PRD/RPT/g' \
        -e 's/REUSE/REUSE SET/g' \
        -e 's/NORESETLOGS/RESETLOGS/' \
        -e 's/OPEN/OPEN RESETLOGS' \
        -e 's/RECOVER DATABASE/RECOVER AUTOMATIC FROM '/oracle/PRT/saparch'
                    UNTIL CHANGE ${SCN} USING BACKUP CONTROLFILE/' \
                                > /oracle/RPT/sapreorg/CONTROL.SQL

# Get current archive log
remsh zhpux5 –n "ls –tr /oracle/PRD/saparch/arch*dbf" | tail –n 4 | \
while read PRDLOG
do
  RPTLOG=$(echo ${PRDLOG} | sed s/prd/rpt/g)
  rcp –p zhpux5:/oracle/PRD/saparch/${PRDLOG} /oracle/RPT/saparch/${RPTLOG}
done
chown orarpt:dba /oracle/RPT/saparch/arch*dbf

# Delete old redo log files
rm /oracle/RPT/*log*/*
```

checks for new `/oracle/<SID>/sapdata` file systems and mounts them if necessary. In order to re-create the control files on the target host, `ora_ux.sh` also regenerates a control trace file on the production host. This picks up any changes that have been made to the layout of the production database. In the control trace file, all occurrences of the old `<SID>` are replaced with the new `<SID>`.

At this point, the UNIX processing is completed, and the new database is available for use. The next step is to refresh and clean up the database:

Listing 6: Script "inf_refresh.sh" Calls "inf_refresh.sql" to Refresh the Informix Database

```
#!/usr/bin/sh
#
# inf_refresh.sh
#
# Author: Bert Vanstechelman
#
# Refresh & rename Informix database

exec > /tmp/inf_refresh.log.$$ 2>&1

# re-start the database

oninit
sleep 180

# rename database
echo "rename database PRD to RPT;"|dbaccess sysutils

# clean-up database
dbaccess RPT inf_refresh.sql
```

Listing 7: Script "inf_refresh.sql"

```
--
-- inf_refresh.sql
--
-- Author: Bert Vanstechelman
--
-- Refresh & rename Informix database

drop table sapuser;
create table rptadm.sapuser (userid char(25) not null,
   passwd char(25) not null) in psapstab lock mode row;
-- passwd should be same as OS passwd for sapr3
insert into sapuser values ('sapr3','sapr3');

grant connect to rptadm;
```

(continued on next page)

9. `RPT_refresh.sh` calls a script to rename the database and replace all the post-processing actions described in the "R/3 Homogeneous System Copy Manual" with SQL scripts.

   - For Informix, this script is `inf_refresh.sh` (called on line 85), which in turn calls `inf_refresh.sql` (see **Listing 6** and **Listing 7**).

(continued from previous page)

```
-- update print server (transaction SPAD)

update sapr3.tsp03c set (pamsserver) = ('zhpux31_RPT_05');
update sapr3.tsp03d set (pamsserver) = ('zhpux31_RPT_05');

-- delete the database statistics (transaction ST03)
delete from sapr3.DBSTATTINF;

-- delete performance statistics (transaction ST03)
delete from sapr3.MONI;
delete from sapr3.PAHI;
delete from sapr3.OSMON;
delete from sapr3.DBSNP;
delete from sapr3.SDBAH;
delete from sapr3.SDBAD;
delete from sapr3.SDBAP;
delete from sapr3.SDBAR;

-- delete the default, start and instance profile (transaction RZ10)
delete from sapr3.TPFET;
delete from sapr3.TPFHT;

-- delete the table used for buffer synchronization (transaction SE16)
delete from sapr3.DDLOG;

-- delete the logon groups (transaction SMLG)
delete from sapr3.RZLLITAB;

-- simulate configuration of Workbench Organizer (SE06)
delete from sapr3.TLOCK;
update tadir set SRCSYSTEM='RPT'
where pgmid='HEAD' and object='SYST';

-- delete all scheduled jobs (transaction SM37 or job RSBTCDEL)
delete from  sapr3.tbtco
where jobname not like 'EU%' and
jobname not like 'RDDIMP%' and
jobname not like 'SAP%' and
jobname not like 'SYST%' and
jobname not like 'ZCLEANUP';

delete from sapr3.tbtcs where jobname not like 'EU%' and
jobname not like 'RDDIMP%' and
jobname not like 'ZCLEANUP' and
jobname not like 'SYST%' and
jobname not like 'SAP%';

delete from sapr3.tbtcp where jobname not like 'EU%' and
jobname not like 'RDDIMP%' and
jobname not like 'SYST%' and
jobname not like 'ZCLEANUP' and
jobname not like 'SAP%';

delete from sapr3.btcevtjob where jobname not like 'EU%' and
jobname not like 'RDDIMP%' and
jobname not like 'ZCLEANUP' and
jobname not like 'SYST%' and
jobname not like 'SAP%';

-- update tables used for the logical output system (transtion SPAD)
update sapr3.tsproms set (startinst, startcmd) = ('zhpux31_RPT_05',
'/utils/dazel/bin/dzl_cb_config -E zhpux31_RPT_05 -R DAZEL -S omscb-RPT
-U DZL -P abcdzl -C 300 –O"DAZELNS=DAZEL:bekaert,dazelsrv\"');

update sapr3.tsploms
set (cmdserver, callbtrgt) = (' zhpux31_RPT_05',' zhpux31_RPT_05');
```

Listing 8: Script "ora_refresh.sql" Refreshes the Oracle Database

```
REM ora_refresh.sql
REM
REM Author: Bert Vanstechelman
REM
REM Refresh and rename Oracle database

REM re-start the database


connect internal

REM re-create the control files
@/oracle/RPT/sapreorg/CONTROL.SQL

REM create the OPS$USER
@/oracle/RPT/sapreorg/create-ops.sql

REM update print server (transaction SPAD)
update sapr3.tsp03c set (pamsserver) = ('zhpux31_RPT_05');
update sapr3.tsp03d set (pamsserver) = ('zhpux31_RPT_05');
```

(continued on next page)

- For Oracle, it is `ora_refresh.sql`, called on line 87 (**Listing 8**).

The SQL script that you write will need to perform the same actions that these scripts perform

(continued from previous page)

```
REM delete the database statistics (transaction ST03)
truncate table sapr3.DBSTATTINF";

REM delete performance statistics (transaction ST03)
truncate table sapr3.MONI;
truncate table sapr3.PAHI;
truncate table sapr3.OSMON;
truncate table sapr3.DBSNP;
truncate table sapr3.SDBAH;
truncate table sapr3.SDBAD;
truncate table sapr3.SDBAP;
truncate table sapr3.SDBAR;

REM delete the default, start and instance profile (transaction RZ10)
truncate table sapr3.TPFET;
truncate table sapr3.TPFHT;

REM delete the table used for buffer synchronization (transaction SE16)
truncate table sapr3.DDLOG;

REM delete the logon groups (transaction SMLG)
truncate table sapr3.RZLLITAB;

REM simulate configuration of Workbench Organizer (SE06)
truncate table sapr3.TLOCK;
update tadir set SRCSYSTEM='RPT'
where pgmid='HEAD' and object='SYST';

REM delete all scheduled jobs (transaction SM37 or job RSBTCDEL)
delete from sapr3.tbtco
where jobname not like 'EU%' and
jobname not like 'RDDIMP%' and
jobname not like 'SAP%' and
jobname not like 'SYST%' and
jobname not like 'ZCLEANUP';

delete from sapr3.tbtcs where jobname not like 'EU%' and
jobname not like 'RDDIMP%' and
jobname not like 'ZCLEANUP' and
jobname not like 'SYST%' and
jobname not like 'SAP%';

delete from sapr3.tbtcp where jobname not like 'EU%' and
jobname not like 'RDDIMP%' and
jobname not like 'SYST%' and
jobname not like 'ZCLEANUP' and
jobname not like 'SAP%';

delete from sapr3.btcevtjob where jobname not like 'EU%' and \
jobname not like 'RDDIMP%' and
jobname not like 'ZCLEANUP' and
jobname not like 'SYST%' and
jobname not like 'SAP%';

REM update tables used for the logical output system (transtion SPAD)

update sapr3.tsproms
set (startinst, startcmd) = ('zhpux31_RPT_05','/utils/dazel/bin/
dzl_cb_config -E zhpux31_RPT_05 -R DAZEL
-S omscb-RPT -U DZL -P abcdzl -C 300 -O”DAZELNS=DAZEL:bekaert,dazelsrv\"');

update sapr3.tsploms
set (cmdserver, callbtrgt) = (' zhpux31_RPT_05',' zhpux31_RPT_05');
```

Listing 9: Create OPS$ User in Oracle

```
REM Oracle create-ops.sql script
REM
REM Author: Bert Vanstechelman
REM
REM Re-create the OPS$<OS_USER>


drop user "OPS$PRDADM" cascade;
create user "OPS$RPTADM" default tablespace PSAPUSER1D
temporary tablespace psaptemp identified externally;
grant connect, resource to "OPS$RPTADM";
create table "OPS$RPTADM".SAPUSER
(USERID VARCHAR2(256), PASSWD VARCHAR2(256));
insert into "OPS$RPTADM".SAPUSER values ('sapr3'.'sapr3');
drop public synonym sapuser;
create public synonym SAPUSER for "OPS$RPTADM.SAPUSER;
```

in their respective environments (I provide some tips on writing this SQL script at the end of the list), namely:

- Rename the database. To rename the database in an Informix environment, connect as `Informix` and use the `rename` statement. In an Oracle environment, run the `CONTROL.SQL` script. The `CONTROL.SQL` script will re-create the control files for the new database. (Note that the `CONTROL.SQL` script can be generated on the host environment using the `alter database backup controlfile to trace`. In the example, we generated it in the `ora_ux.sh` script.)

- For Informix, grant the new user `<SID>adm` (this user was copied in step 7) connect and select permissions to the new database. For Oracle, create a new `OPS$<OSUSER>` and grant sufficient permissions (see the `create-ops.sql` script in **Listing 9**).

- Delete all database statistical information. For Informix, delete all entries in table DBSTATTINF. For Oracle, delete all entries in tables DBSTAIHORA, DBSTATHORA, DBSTATIORA, and DBSTATTORA. These tables contain the statistical information used by SAP in transactions ST03, ST04, and DB13.

- Delete all performance statistics in tables MONI, PAHI, OSMON, DBSNP, SDBAH, SDBAD, SDBAP, and SDBAR.

- Delete all jobs and job steps that might conflict with production. This is a very important step. If it fails, production jobs will run on your target system as soon as it starts up. If these jobs generate output, it might confuse your users.

- Delete everything in the following four tables: TBTCO (contains all jobs); TBTCS (jobs are scheduled via this table); TBTCP (the individual job steps are in this table); BTCEVTJOB (contains all jobs waiting for an event). You can leave the standard SAP jobs. You can also selectively delete jobs if you know which jobs generate output.

- Update the print server. Update field PAMSERVER in table TSP03C and TSP03D with the new HOSTNAME_SID_INSTANCE_NUMBER. If you are using a Logical Output Management System such as DAZEL, update the STARTINST and STARTCMD entries in table TSPROMS. Update table TSPLOMS with the new HOSTNAME_SID_INSTANCE_NUMBER for the callback service.

- Delete all the old profiles in the database. Delete all entries in tables TPFET and TPFHT. There is no need to import the new profiles. The database is overwritten every day anyway.

- Delete all entries in table DDLOG. It is used to synchronize the buffers between the application servers.

- If the new system is to be actively used in an SAP landscape, use the Workbench Organizer (transaction SE06) to configure your system and add it to the transport routes with the Transport Management System (transaction STMS).

  If it is going to be used as a standalone system, just delete all entries in table TLOCK and update the SID in table TADIR. Table TLOCK contains all locks of all transports, and it contains repairs and customizing requests that have not been released. This is a quick and dirty solution, but it gets the job done.

- Delete all the logon groups and assignments. Delete all entries in table RZZLITAB.

✓ Tip

All SAP data is in tables. If you know where to find what you need, you can easily write SQL scripts that update the tables directly, just as they would be updated from within SAP. Fortunately, the information you require is easy to find.

***Figure 5** is a list of the tables to manipulate. In the SAP GUI, execute the transaction and press F1, go to the technical information screen, and a window will pop up telling you what you need to know. This window, together with the ABAP Data Dictionary, gives you all the information you require on the tables and their specific fields.*

10. The SAP system has been renamed, cleaned, and is ready to be started on the target host, along with the SAP housekeeping jobs. On line 89 in Listing 2, RPT_refresh.sh calls the post-processing script (sappost.sh, shown in

Figure 5 *SAP Tables That Need To Be Manipulated*

Table	Contents	Action
BTCEVTJOB	List of jobs waiting for events	Delete scheduled user jobs (SM37)
DBSNP	Database snapshots	Delete all entries (SE11)
DBSTATTINF	Informix update statistical information	Delete all entries (SE11)
DDLOG	Buffer synchronization	Delete all entries (no SAP transaction); use TRUNCATE/DELETE
MONI	Monitor table MONI	Delete all entries (SE11)
OSMON	Operating system monitoring data	Delete all entries (SE11)
PAHI	History of system, DB, and SAP parameter	Delete all entries (SE11)
RZLLITAB	Assignments of logon/server groups to instances	Delete all entries (SMLG)
SDBAD	Detail table for DBA logs	Delete all entries (SE11)
SDBAH	Header table for DBA logs	Delete all entries (SE11)
SDBAP	DBA planning table	Delete all entries (SE11)
SDBAR	DBA resource table	Delete all entries (SE11)
TBTCO	Job status overview table	Delete scheduled user jobs (SM37)
TBTCP	Batch job step overview	Delete scheduled user jobs (SM37)
TBTCS	Batch schedule table	Delete scheduled user jobs (SM37)
TLOCK	Change and Transport System: lock table	Configure Workbench Organizer (SE06)
TPFET	Table of profile parameters	Delete all entries (RZ10)
TPFHT	Profile header, administration data for profiles in DB	Delete all entries (RZ10)
TSP03C	Spool: device description extension	Update print server (SPAD)
TSP03D	Spool: device description (new format)	Update print server (SPAD)
TSPLOMS	Logical Output Management Systems for access method “E”	Change OMS configuration for Dazel (SPAD)
TSPROMS	Real Output Management Systems for access method “E”	Change OMS configuration for Dazel (SPAD)

Listing 10: Install the New SAP License, Start SAP, and Trigger Cleanup

```
#!/usr/bin/sh
#
# sappost.sh
#
# Author: Bert Vanstechelman
#
#

exec > /tmp/sappost.log.$$ 2>&1

date

# SAP (insert SAP license and re-start SAP)
saplicense -install ifile=/home/${LOGNAME}/newlicense.dat TRACE=2

${SAP_start}

sleep 120

# start sap cleaning jobs
/usr/sap/${SAP_SID}/SYS/exe/run/sapevt ZCLEANUP_EVENT name=${SAP_SID} nr=10
```

Figure 6 *Steps in SAP Cleanup Job*

Program	Variant	Action
RSBTCDEL	ZJOBDEL_1	Deletes finished and canceled jobs
RSBDCREO	ZBATCH_1	Deletes batch input logs
RSSNAPDL	ZABAP_1	Deletes ABAP short dumps
RSBPSTDE	ZJOBSTATS_1	Reorganizes job statistics
RSPO1043	ZTEMSE_1	Checks and cleans TEMSE
RSPO0041	ZSPOOL_1	Deletes all old spool jobs

Listing 10). This script starts SAP, installs the new license, and sends SAP an event to trigger SAP housekeeping jobs (listed in **Figure 6**).

When you write your own script, I recommend running all the SAP housekeeping jobs with a variant that cleans up as much as possible. You can do this by creating a new job (on the source production instance) that includes all SAP standard cleaning jobs. The job should start on the target host when the system receives an event. (You need to create the job on the source production instance, because creating it on the target is not very useful, as the database is overwritten with every refresh. However, you should *never* execute it on the source instance, because it would delete data not useful on the copied database, but essential to the production host.)

Here are the steps:

1. Create new variants for the standard SAP jobs.
2. Go to the ABAP Workbench (transaction SE38) and define new variants for the standard cleanup jobs. Give your variants the highest cleanup values possible.
3. Create a new SAP event (for example, `ZCLEANUP_EVENT`) in the Maintain Event window (transaction SM62).
4. In the SAP menu go to **Tools** → **CCMS** → **Jobs** → **Definition** (transaction SM36). Create a new job (job class A), with the variants created in the previous step and the following SAP programs: `RSBTCDEL`, `RSBDCREO`, `RSSNAPDL`, `RSBPSTDE`, `RSPO1043`, and `RSPO0041`. The job should start after event `ZCLEANUP_EVENT`.
5. Using an ASCII file (`newlicense.dat`, shown in **Listing 11**) as a parameter, install the new license automatically with `saplicense`. Don't ask why, but you need to add the parameter `TRACE=2` to get it to work. The ASCII file can be generated in OSS.

Listing 11: ASCII File with SAP License

```
SAPSYSTEM = RPT
CUSTOMER-KEY = B1790751340
INSTNO = 1620001306
EXPIRATION = 99991231
LKEY = RPOUMPXPSQOVGITMNL8E7QAC
```

Helpful Hints

Setting up an environment to support an automatic, homogeneous system copy is time-consuming, and definitely not easy. The good news is that it's a one-time effort that yields a big benefit — a system copy of your production system that is easily kept up-to-date. Here are a few hints for a successful project:

✓ **Be well prepared before you start.** You might need to upgrade your storage disk array to support the disk copy. You might even need to acquire extra disk space. You should certainly get some get hands-on experience performing a disk copy with the API (EMC^2's Solution Enabler).

✓ **Proceed with care.** Although you are not changing anything on your production system, there are other things that you may need to take into consideration. For example, when the copy of your production system comes up for the first time, it might generate output, which can be confusing for your users, or it might try to connect to external systems.

✓ **Don't get frustrated if something goes less quickly than expected.** There is a lot you need to take into consideration: hardware, operating system, database, and SAP itself. Take it step by step, and don't be afraid to restart if you're completely stuck. It took me some time before my first automatic system copy was working smoothly. (I built it step by step, adding to it as new ideas came along. The only tricky part involved background jobs. They started running on the reporting host and generating output, which confused users, especially in the production plants.)

Conclusion

SAP databases have a tendency to grow and grow and grow. The bigger the database, the more difficult and time-consuming it becomes to make a copy. Fortunately, a lot of customers are switching to integrated disk-array solutions for their ever-growing SAP landscapes. These new high-tech environments offer great opportunities for implementing the automatic system copy procedure, because they make it possible to quickly copy large amounts of data without much impact on the production host.

Setting up an automatic homogeneous system copy procedure is time-consuming, and it needs to be well prepared. But once it's implemented, end users, consultants, ABAP developers, and, of course, Basis administrators can greatly benefit.

The procedure presented in this article can be used to create systems for all kinds of purposes: to support upgrades, a Euro conversion, to set up a special training system, or to refresh quality assurance systems on a regular basis. Although only Informix and Oracle were discussed, one can easily adapt the concept for other databases like DB2 or SQL Server. I'm sure that many of you Basis administrators are already thinking about how this procedure might be a solution for your own environment.

For a listing of helpful references on the technologies used in this solution, and the OSS notes used, visit the "Download Files" section at **www.SAPpro.com**.

Bert Vanstechelman studied computer science at the BME Institute in Ghent, Belgium. He works as an independent SAP Basis consultant and has approximately six years of professional computing experience. His most recent long-term assignments have been in SAP Basis consulting roles with Oracle and Informix databases in both UNIX and NT environments. He specializes in SAP hardware infrastructures, high availability, installations, upgrades, backup/recovery issues, and last, but not least, OS/DB migrations. He can be reached at Bert_Vanstechelman@non.hp.com or bert.vanstechelman@skynet.be.