



Technical Report

# SAP System Copy with SnapManager for SAP

SAP Competence Center, NetApp  
February 2010 | TR-3812

## **ABAP, JAVA, OR DOUBLE STACK**

This technical report describes the process of creating an SAP system copy with SnapManager® for SAP® (SMSAP) for SAP systems with an ABAP, a Java™, or a double stack.

## TABLE OF CONTENTS

<b>1</b>	<b>OVERVIEW .....</b>	<b>3</b>
1.1	SAP SYSTEM COPIES .....	3
<b>2</b>	<b>OVERVIEW SAP SYSTEM COPY WITH SNAPMANAGER FOR SAP .....</b>	<b>6</b>
2.1	SAP SYSTEM COPY JAVA AND ABAP STACK.....	7
2.2	SAP SYSTEM COPY ABAP STACK ONLY .....	10
<b>3</b>	<b>SAP SYSTEM COPY JAVA OR DOUBLE STACK .....</b>	<b>13</b>
3.1	TASKS ON THE SOURCE SYSTEM.....	13
3.2	NEW INSTALLATION OF THE TARGET SYSTEM.....	16
3.3	REFRESH OF TARGET SYSTEM.....	22
<b>4</b>	<b>SAP SYSTEM COPY ABAP STACK ONLY .....</b>	<b>26</b>
4.1	TASKS ON THE SOURCE SYSTEM.....	26
4.2	NEW INSTALLATION OF THE TARGET SYSTEM.....	26
4.3	REFRESH OF TARGET SYSTEM.....	29
<b>5</b>	<b>SETUP OVERVIEW .....</b>	<b>31</b>

# 1 OVERVIEW

## 1.1 SAP SYSTEM COPIES

### CAPACITY REQUIREMENTS

When creating SAP system copies with most storage architectures, space must be allocated to accommodate the entire size of the source database. This can drastically increase the amount of storage required to support a single production SAP instance.

During a typical project a 1TB SAP production system will be copied to a quality assurance (QA) system, a test system, and a training system. With conventional storage architectures, this requires an additional 3TB of storage. Furthermore, it requires a significant amount of time to first back up the source system and then to restore the data to the three target systems.

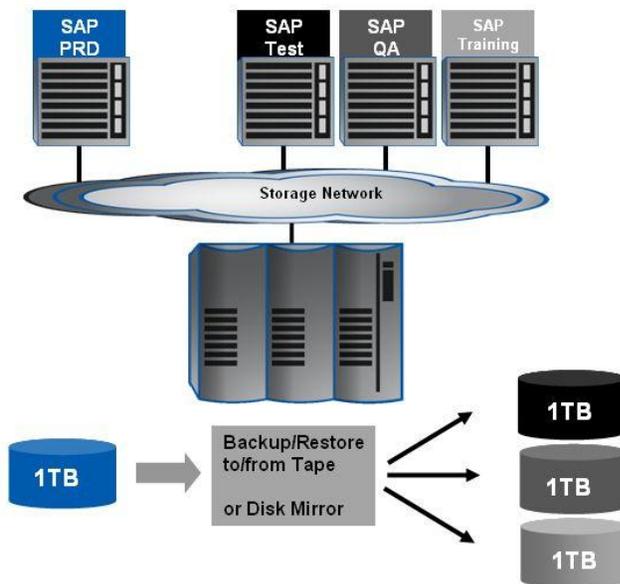


Figure 1) SAP system copy: standard approach.

In contrast, when using NetApp® FlexClone® technology to create SAP system copies, only a fraction of the storage space is required. NetApp FlexClone technology uses Snapshot™ copies, which are created in a few seconds without interrupting the operation on the source system, to perform SAP system copies. Because the data is not copied but referenced in place, the amount of storage required is limited to only data that is changed at the source and the target system and therefore significantly decreases the disk space needed for SAP system copies.

As a result, the capacity requirements for a system copy in a NetApp storage environment depend on the refresh cycle of the target systems. As longer test systems are kept, more block changes will happen from the source and the target system. Storage requirements also depend on the number of copies that are made from the same source. Of course more copies of the same source system will result in higher storage savings.

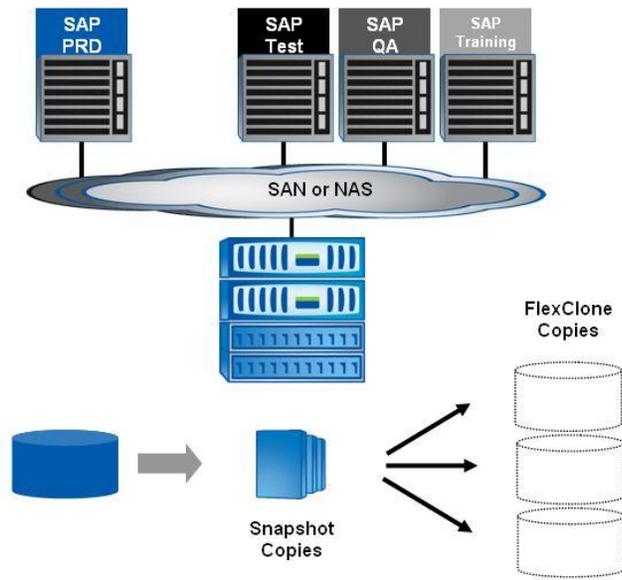


Figure 2) SAP system copy: NetApp approach at primary storage.

On the source system a database-consistent Snapshot copy of the data files is created. This is done during online operation and has no performance impact on the source system. Therefore this step can be carried out at any time.

The FlexClone copy can be created at the same storage system or at a secondary storage system.

The secondary storage system could already be in place and used as a disk-to-disk backup device using SnapVault® or a disaster recovery solution using SnapMirror®. The backup or disaster recovery replication images can be accessed for reading and writing using FlexClone technology. Existing backup or disaster recovery images will be utilized for test environments, turning expenses into assets. As a side effect the backup and recovery or disaster recovery solution is tested without any additional effort and without any interruption.

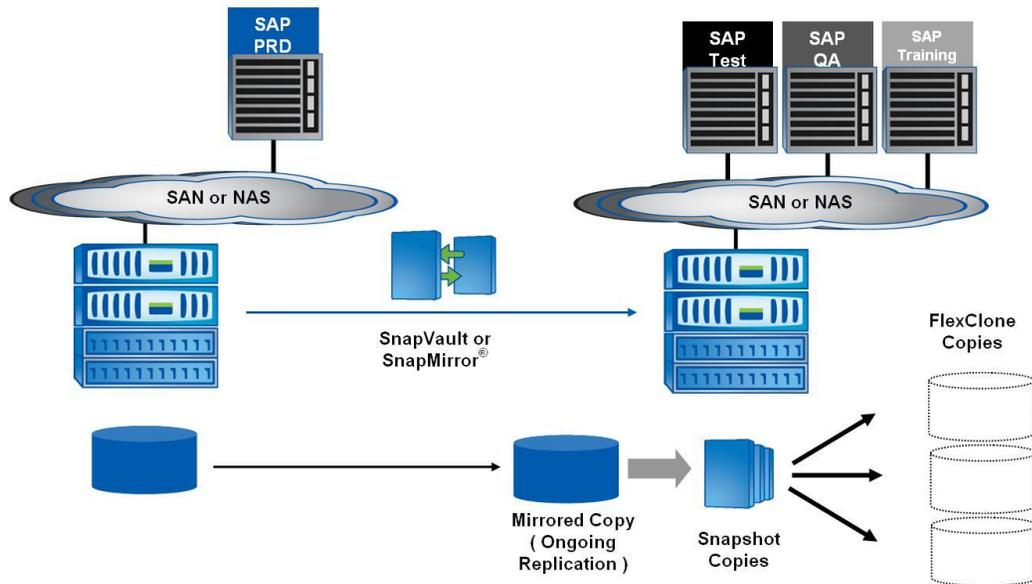


Figure 3) SAP system copy: NetApp approach at secondary storage.

## TIME REQUIREMENTS

The time required to create an SAP system copy can be subdivided into three parts:

- Time to restore the backup to the target system
- Time to perform OS and database-specific postprocessing
- Time to perform SAP application postprocessing  
SAP postprocessing depends on the customer's SAP environment. Some customers can finish postprocessing in a few hours while other customers need several days to accomplish this task.

In a conventional system copy process, the data is backed up to tape and then restored, which takes a great deal of time. If an online backup is used, there is no downtime for the source system; however, there might be a performance impact on the source system during the backup. Because of the large number of logs that need to be applied, the time required to recover the database and make it consistent is greatly increased, possibly adding hours to the system copy process. If an offline backup is used, the source system is shut down, resulting in a loss of productivity.

The figures below show an example describing the difference between the amount of time spent creating an SAP system copy with NetApp storage versus the time spent using a conventional approach.

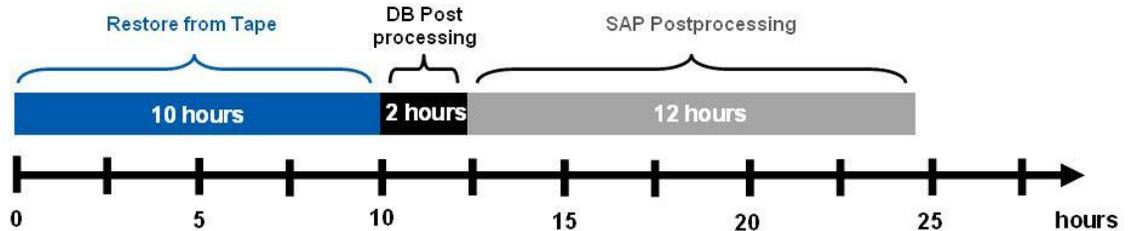


Figure 4) SAP system copy: standard approach.

All steps up to the point when the SAP system can be started on the target host can be accomplished in a few minutes using the NetApp solution compared to several hours with the standard approach. With both approaches the SAP postprocessing needs to be done as an additional step.

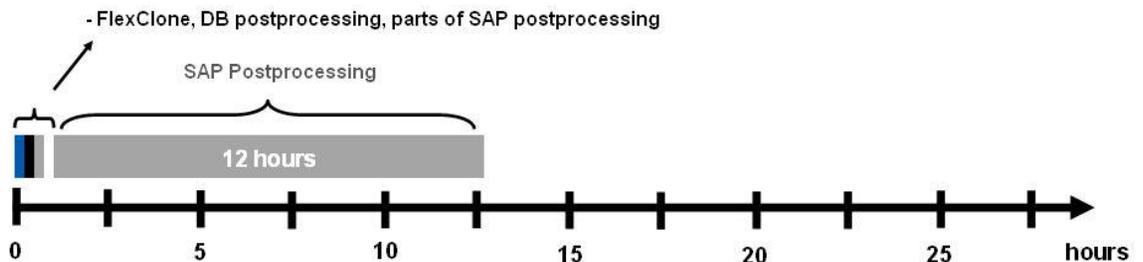


Figure 5) SAP system copy: NetApp approach.

A key requirement to successfully managing an SAP environment is the ability to create copies of production data for use in testing, quality assurance, or training. NetApp Snapshot and FlexClone technologies allow a fast and space-efficient creation of SAP systems.

## 2 OVERVIEW SAP SYSTEM COPY WITH SNAPMANAGER FOR SAP

This technical report explains how SnapManager for SAP and the underlying Snapshot and FlexClone features can be used together with SAPinst to create an SAP system copy.

The document also describes how SAPinst can be used with minimal or no required user input in order to accelerate the system copy process.

Within the paper two scenarios are covered:

- **New installation**  
New setup of the target system based on a database copy of the source system
- **System Refresh**  
Refresh of the database of the target system based on a database copy of the source system

The following table shows the differences between using SAPinst alone and using SAPinst together with SMSAP.

Table1) System copy overview.

	ABAP or Java stack	New Installation		System Refresh	
		SAPinst alone	SAPinst with SMSAP	SAPinst alone	SAPinst with SMSAP
SAPinst needs to run on source system	ABAP	No	No	No	No
	Java	Yes	Yes	Yes	Yes
yes	ABAP	Yes	No	Yes	No
	Java	Yes	No	Yes	No
Backup of source system	ABAP	Offline	Online	Offline	Online
	Java	Offline	Online	Offline	Online
SAPinst needs to run on target system	ABAP	Yes	Yes	Yes	No
	Java	Yes	Yes	Yes	Yes

The main advantages of SMSAP are:

- Snapshot-based online backups are used as a base for the system copy. The Snapshot backup can be done at any time and has no impact on the production source system.
- A FlexClone copy is used to attach the online Snapshot backup of the source system to the target system. The FlexClone copy is done in a few minutes.
- SMSAP handles all database-specific tasks of the system copy. Therefore there is no need to run OraBrCopy any more.
- For systems with ABAP stack only a system refresh is done without the need to run SAPinst.

## 2.1 SAP SYSTEM COPY JAVA AND ABAP STACK

### SAP STANDARD APPROACH

The following figure gives an overview of the system copy process of an SAP system with double stack or Java stack only. This process is described in the SAP guide “System Copies for SAP Systems Based on SAP Netweaver 7.0 SR3 ABAP and Java” in the chapter “Database-Specific System Copy.”

There are several steps that need to be executed on the source system:

- Background jobs need to be set to “scheduled” in order to avoid starting these jobs on the target system. Since SAPinst will start the SAP system on the target server during the installation the background jobs need to be handled on the source system.
- SAPinst is used to archive the Software Deployment Manager (SDM) data.
- The OraBrCopy tool is used to create a “control file to trace.”
- An offline backup needs to be created.  
SAPinst only supports offline backups as a source for the system copy.

The SDM archive and the CONTROL.SQL file are required for the SAPinst process on the target system.

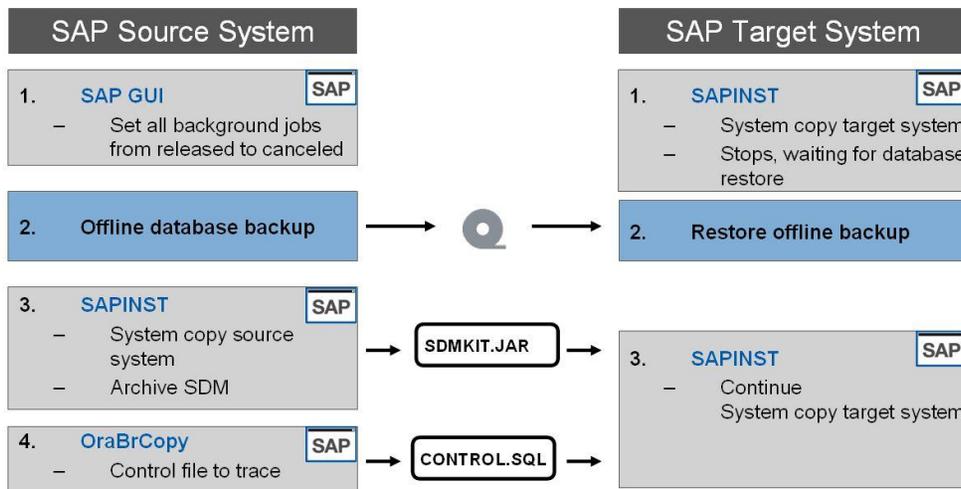


Figure 6) SAP system copy double stack, standard approach.

The SAP standard approach includes many manual tasks and can't be automated.

## SYSTEM COPY WITH SMSAP – NEW INSTALLATION

The approach for creating an SAP system copy for a new installation with SMSAP differs in the following tasks:

- There is no need to run OraBrCopy on the source system, since SMSAP handles the change of the database SID on the target host.
- Online Snapshot backups can be used as the source of the system copy, because SMSAP handles the recovery of the online backup on the target host. SAPinst won't see a difference.
- There is no need to handle the background jobs on the source system before creating the Snapshot backup. The handling of the background jobs is done within a postcloning plug-in of SMSAP.

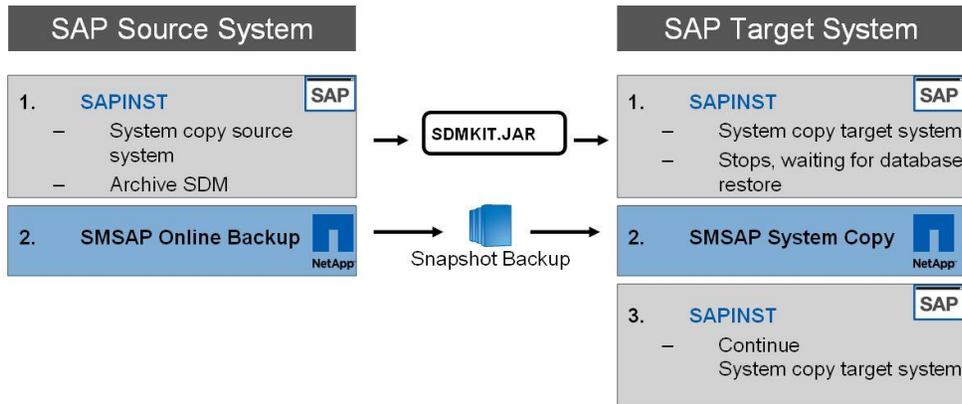


Figure 7) SAP system copy double stack, new installation, NetApp approach.

During the new installation of the target system all required input and configuration should be done and prepared in order to be able to accelerate and automate any further refresh of the target system.

- Preparation of “unattended mode” for SAPinst on the source system  
SAPinst will run without any user input.
- Preparation of SAPinst input files on the target host  
SAPinst will run with minimal user input.
- Preparation of cloning specification for SMSAP  
SMSAP will run in unattended mode.

The installation includes the following steps:

- (Source)  
Run SAPinst to archive SDM data.  
Preparation of unattended mode for all subsequent runs.
- (Source)  
Run SMSAP to create online backup of the SAP system.
- (Target)  
Install operating system; prepare file systems for SAP installation; install SnapDrive® and SMSAP.
- (Target)  
Run SAPinst with system copy service.  
Preparation of all input for subsequent runs.  
SAPinst will stop for the restore of the database.
- (Target)  
Run SMSAP to create a clone of the database.  
During this step the cloning specification parameter can be saved in a file for subsequent runs.

- (Target)  
Continue SAPinst.

### SYSTEM COPY WITH SMSAP – REFRESH OF TARGET SYSTEM

A refresh of the target system includes the following steps:

- (Target)  
Stop SAP system only. Database needs to run when clone is deleted with SMSAP in the next step.
- (Target)  
Delete clone using SMSAP.
- (Source)  
Create online backup of source system with SMSAP (optional).
- (Target)  
Create clone using SMSAP and the Snapshot copy created before or any existing Snapshot copy. The SMSAP cloning process uses the clone specification that was created during the new installation of the target system as described in the previous section. When SMSAP has finished the cloning process the database on the target system is up and running with the changed SID. The background jobs are already canceled by an SQL<sup>®</sup> script that is executed as a postcloning plug-in of SMSAP.
- (Source)  
Run SAPinst on the source system in “unattended mode” using the environment that has been prepared during the new installation of the target system.
- (Target)  
Run SAPinst on the target system using the environment that has been prepared during the new installation of the target system. Only minimal user input is required. When SAPinst stops to ask for the restore of the database it can be continued immediately, since the database was made available by SMSAP.

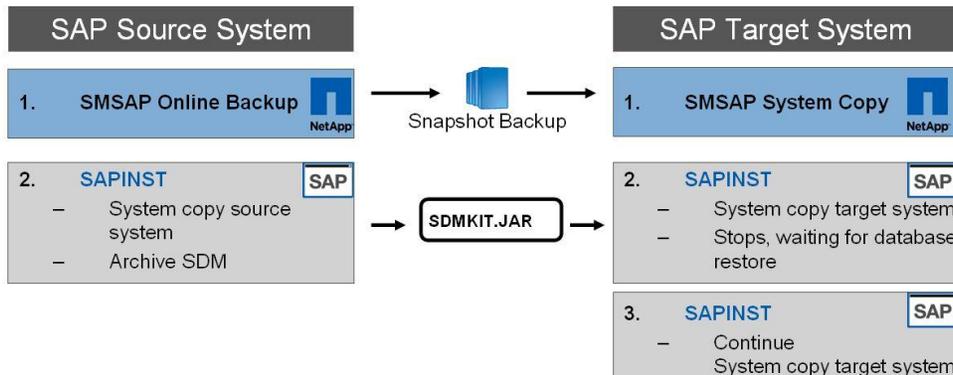


Figure 8) SAP system copy double stack, system refresh, NetApp approach.

Refreshing the target system only requires minimal user input when running SAPinst on the target system. All the other tasks can run unattended.

The overall run time is around 45 minutes in the described setup.

## 2.2 SAP SYSTEM COPY ABAP STACK ONLY

### SAP STANDARD APPROACH

The following figure gives an overview of the system copy process of an SAP system with ABAP stack only. This process is described in the SAP guide “System Copies for SAP Systems Based on SAP Netweaver 7.0 SR3 ABAP” in the section “Database-Specific System Copy.”

There are several steps that need to be executed on the source system:

- Background jobs need to be set to “scheduled” in order to avoid these jobs getting started on the target system. Since SAPinst will start the SAP system on the target server during the installation the background jobs need to be handled on the source system.
- The OraBrCopy tool is used to create a “control file to trace.”
- An offline backup needs to be created.  
SAPinst only supports offline backups as a source for the system copy.

The CONTROL.SQL file needs to be made available to the SAPinst process on the target system.

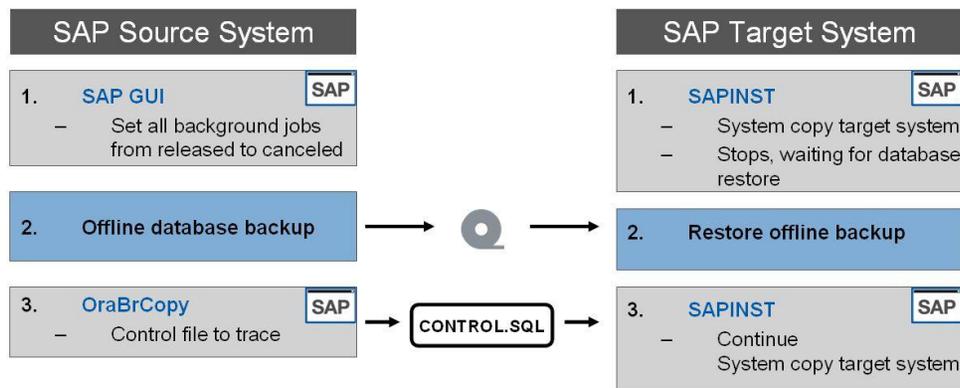


Figure 9) SAP system copy ABAP stack, standard approach.

The SAP standard approach includes many manual tasks and can't be automated.

## SYSTEM COPY WITH SMSAP – NEW INSTALLATION

The approach of creating a SAP system copy for a new installation with SMSAP differs in the following areas.

- There is no need to run OraBrCopy on the source system, since SMSAP handles the change of the database SID on the target host.
- Online Snapshot backups can be used as the source of the system copy, because SMSAP handles the recovery of the online backup on the target host. SAPinst won't see a difference.
- There is no need to handle the background jobs on the source system before creating the Snapshot backup. Background job handling is done within a postcloning plug-in of SMSAP.

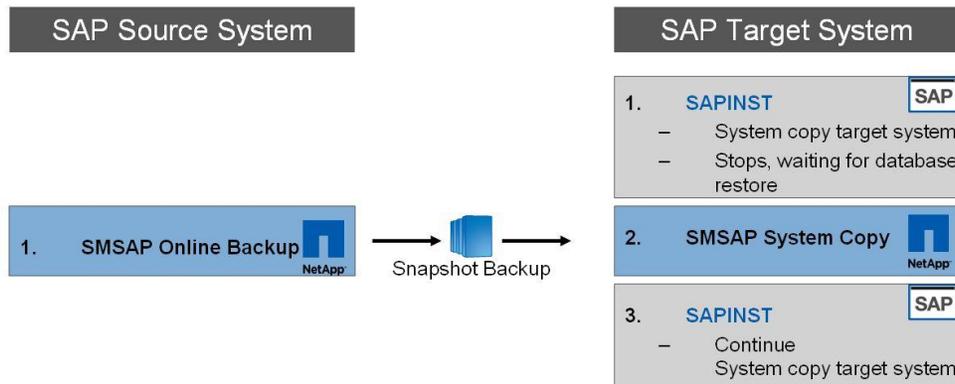


Figure 10) SAP system copy ABAP stack, new installation, NetApp approach.

Installation includes the following steps:

- (Source)  
Run SMSAP to create online backup of the SAP system.
- (Target)  
Install operating system; prepare file systems for SAP installation; install SnapDrive and SMSAP.
- (Target)  
Run SAPinst with task system copy service.
- (Target)  
Run SMSAP to create a clone of the database. During this step the cloning specification parameters can be saved in a file for subsequent runs.
- (Target)  
Continue SAPinst.

## SYSTEM COPY WITH SMSAP – REFRESH OF TARGET SYSTEM

A refresh of the target system will run with the following steps.

- (Target)  
Stop SAP system only. Database needs to run when clone is deleted with SMSAP in the next step.
- (Target)  
Delete clone using SMSAP.
- (Source)  
Create online backup of source system with SMSAP (optional).
- (Target)  
Create clone using SMSAP and the Snapshot created before or any existing Snapshot. The SMSAP cloning process uses the clone specification that was created during the new installation of the target system as described in the previous chapter. When SMSAP has finished the cloning process the database on the target system is up and running with the changed SID. The background jobs were already canceled by an SQL script that is executed as a postcloning plug-in of SMSAP.

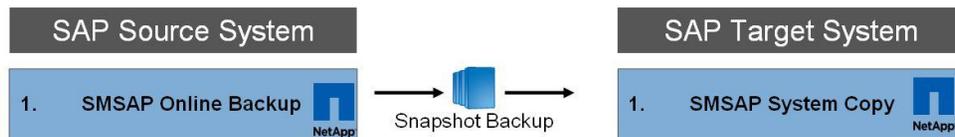


Figure 11) SAP system copy ABAP stack, system refresh, NetApp approach.

Refreshing the target system requires no user input. All tasks can run unattended.

The overall run time is less than 20 minutes in the described setup.

## 3 SAP SYSTEM COPY JAVA OR DOUBLE STACK

### 3.1 TASKS ON THE SOURCE SYSTEM

When a system copy of a Java or double stack system is required, it is always necessary to run SAPinst on the source system to archive the Software Deployment Manager (SDM) data if the SDM data has changed.

This section describes how SAPinst can be prepared so that all subsequent runs can be done using the unattended mode of SAPinst.

#### ORABRCOPY TOOL

There is no need to run the OraBrCopy tool on the source system, since SMSAP handles the change of the SID on the target system. A generic CONTROL.SQL is used and needs to be available when SAPinst is executed on the target host. The file must have the following content.

```
bash-3.00# cat CONTROL.SQL

REM =====
REM CONTROL.SQL
REM =====

STARTUP
```

#### SAPINST ON THE SOURCE SYSTEM

SAPinst is executed using the homogenous system copy method.

All necessary DVD installation images must be available and accessible from the host where the installation is done.

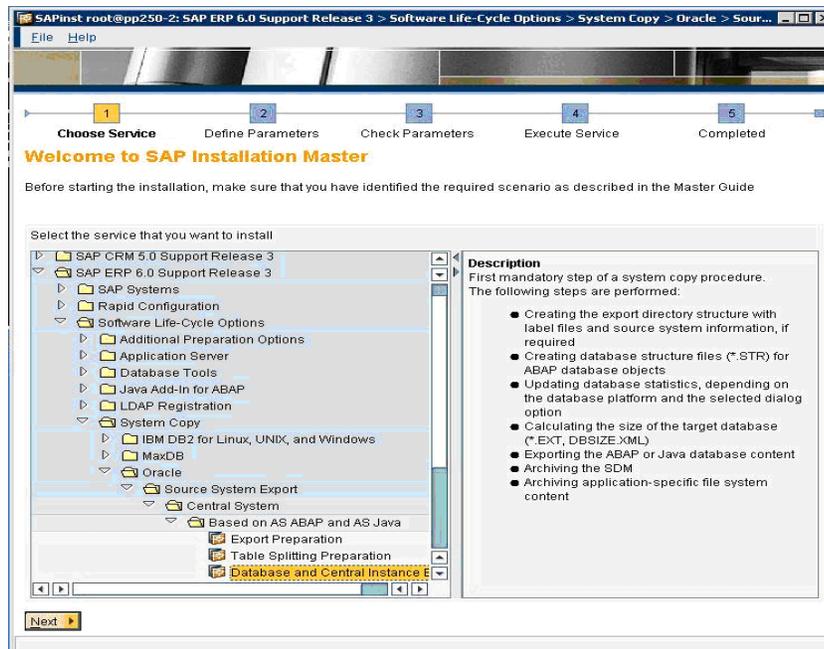


Figure 12) SAPinst—task on the source system.

Several SAPinst installation screens follow.

The export location of the SDM archive should be a shared file system that can also be accessed by the target host. Otherwise the archive has to be copied to the target host before running SAPinst.

In the parameter summary screen of SAPinst, “Use database specific tools” needs to be selected.

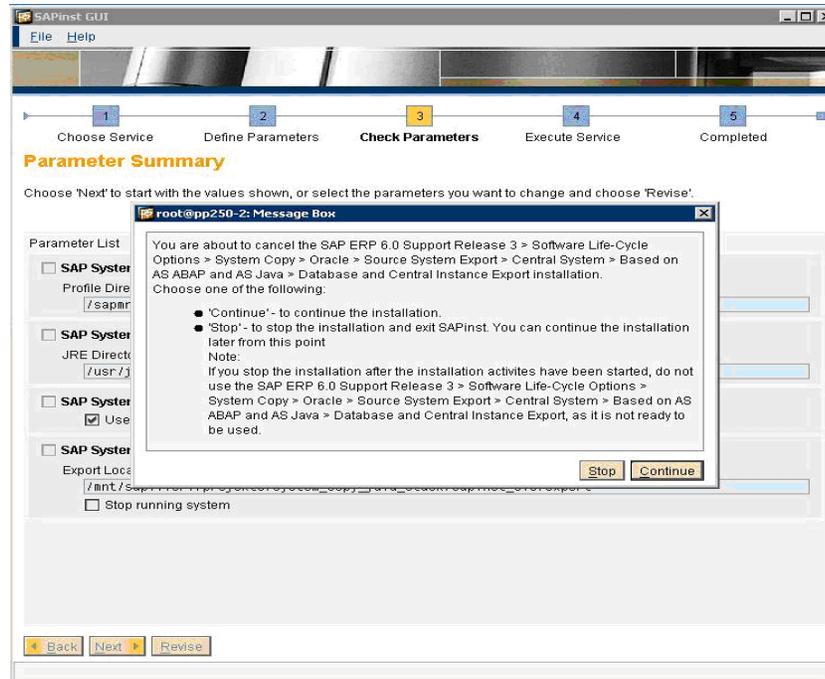


Figure 13) SAPinst—prepare unattended installation.

Now SAPinst is stopped by closing the SAPinst GUI in order to prepare the unattended installation mode. More details can be found in SAP Note 950619. The files doc.dtd, inifile.xml and keydb.dtd from the SAPinst installation directory need to be copied to a “template” directory. The template directory has the following content.

```
bash-3.00# ls -al ./unattended
drwxr-xr-x  3 root    root      4096 Feb 11 15:30 .
drwxrwxr-x  7 root    SAPinst  4096 Feb 11 15:30 ..
-rwxr-x---  1 root    root      3436 Feb 11 15:25 doc.dtd
-rw-r-----  1 root    root      7944 Feb 11 15:25 inifile.xml
-rwxr-x---  1 root    root      1717 Feb 11 15:25 keydb.dtd
-rw-r-----  1 root    root         80 Feb 11 15:27 start_dir.cd
```

In addition the file start\_dir.cd needs to be created manually. This file needs to include a line for each installation DVD that is needed.

```
bash-3.00# cat start_dir.cd
/mnt/sapcds/SAP_ERP_6.0_SR3_Oracle_Solaris/Installation_Master/IM_SOLARIS_SPARC
```

A new SAPinst installation directory needs to be created and all files needs to be copied there.

```

bash-3.00# mkdir SAPinst_source
bash-3.00# chmod 777 SAPinst_source
bash-3.00# chown root:SAPinst SAPinst_source
bash-3.00# cp -pr ./unattended/* ./SAPinst_source

```

Now SAPinst can be started in the new directory and will run in unattended mode.

```

/mnt/sapcds/SAP_ERP_6.0_SR3_Oracle_Solaris/Installation_Master/IM_SOLARIS_SPARC/SAP
inst SAPINST_EXECUTE_PRODUCT_ID=d0e17960 SAPINST_SKIP_DIALOGS=true -nogui -
nogui_server

```

SAP Note 950619 describes how to find the correct SAPINST\_EXECUTE\_PRODUCT\_ID.

After SAPinst has finished, the SDM archive is available in the export directory, which has been given as an input for SAPinst.

```

bash-3.00# pwd
/mnt/sapfiler1/projekte/system_copy_java_stack/SAPinst_S10/export
bash-3.00# ls -al *
-rw-rw-r-- 1 root      SAPinst      5 Feb 12 15:23 LABELIDX.ASC
JAVA:
total 40
drwxrwxr-x 4 s10adm  sapsys      4096 Feb 12 15:23 .
drwxr-xr-x 3 root    root        4096 Feb 20 11:53 ..
drwxrwxr-x 3 s10adm  sapsys      4096 Feb 12 15:25 APPS
-rw-rw-r-- 1 s10adm  sapsys       19 Feb 12 15:23 LABEL.ASC
-rw-rw-r-- 1 s10adm  sapsys       14 Feb 12 15:23 LABELIDX.ASC
drwxrwxr-x 2 s10adm  sapsys      4096 Feb 12 15:25 SDM
-rw-rw-r-- 1 s10adm  sapsys       187 Feb 12 15:23 SOURCE.PROPERTIES
bash-3.00# du -h .
882M  ./JAVA/SDM
 8K   ./JAVA/APPS/ADS
12K   ./JAVA/APPS
882M  ./JAVA
882M  .

```

## 3.2 NEW INSTALLATION OF THE TARGET SYSTEM

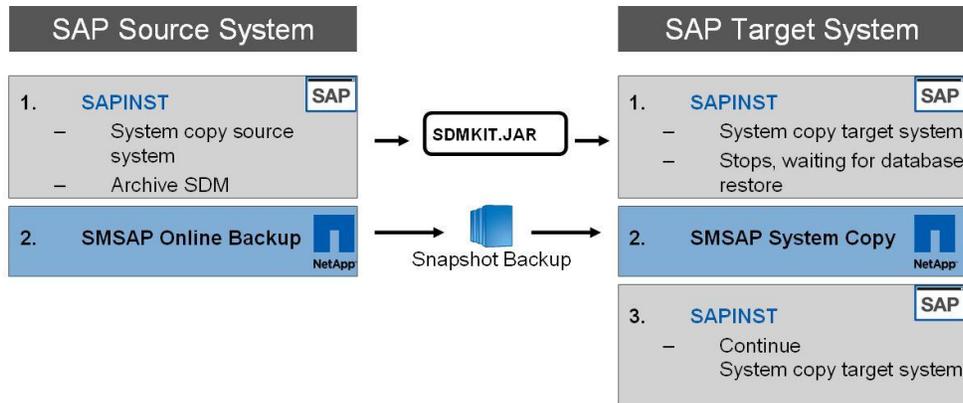


Figure 14) SAP system copy double stack, new installation, NetApp approach.

### BASIC SETUP

All necessary file systems except the sapdata file systems need to be set up as described in the SAP installation guide. The sapdata file systems will be connected later using a FlexClone copy.

#### SDU Installation

- Configuration access to storage controller(s)

```
bash-3.00# snapdrive config set root sapfiler2
Password for root:
Retype password:
bash-3.00#
```

- Configuration of FlexClone for cloning in snapdrive.conf configuration file

```
san-clone-method="optimal" # Clone methods for snap connect
```

#### SMSAP Installation

- No additional configuration is necessary.

### SAPINST ON THE SOURCE SYSTEM

If SDM data has changed, SAPinst needs to run on the source system to archive the current SDM data. SAPinst runs in unattended mode as described in section 3.1. The SDM data will be available in the export location.

## SAPINST ON TARGET SYSTEM

SAPinst is used to install the SAP system using the homogenous system copy method.

All necessary DVD installation images must be available and accessible from the host where the installation is done.

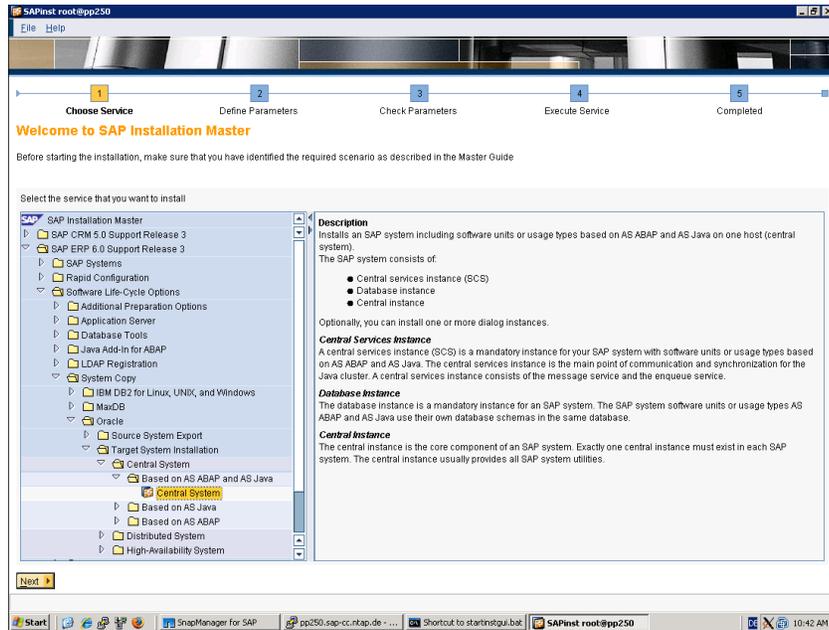


Figure 15) SAPinst—tasks on the target host.

Several SAPinst installation screens follow.

“Backup/Restore installation method” needs to be selected as the “database installation method.”

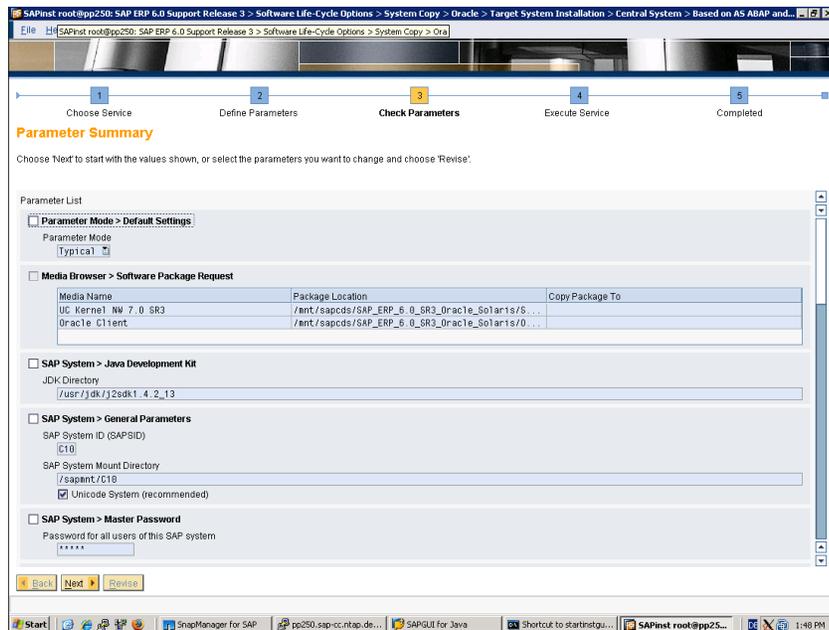


Figure 16) SAPinst—parameter summary.

At the “Parameter Summary” screen the SAPinst GUI needs to be closed to stop the SAPinst process.

This approach allows you to save all the input for future use.

In order to save the input from the SAPinst GUI, copy the SAPinst directory to a “template” directory. (“SAPinst\_Dir” is the directory where SAPinst was executed.)

```
bash-3.00# mkdir golden_template
bash-3.00# chmod 777 golden_template/
bash-3.00# chown root:SAPinst golden_template/
bash-3.00# cp -pr ./SAPinst_Dir/* ./golden_template/
```

The generic CONTROL.SQL, as described in section 3.1, needs to be copied to the template directory.

A new directory is created to run SAPinst and all data from the template directory is copied to this new directory.

```
bash-3.00# mkdir New_SAPinst_Dir
bash-3.00# chmod 777 New_SAPinst_Dir
bash-3.00# chown root:SAPinst New_SAPinst_Dir
bash-3.00# cp -pr ./golden_template/* ./New_SAPinst_Dir
```

SAPinst can now be started in the new directory.

```
bash-3.00#
mnt/sapcds/SAP_ERP_6.0_SR3_Oracle_Solaris/Installation_Master/IM_SOLARIS_SPARC/SAPinst -nogui
```

SAPinst now skips all the parameter input and the execution of the service can be started immediately.

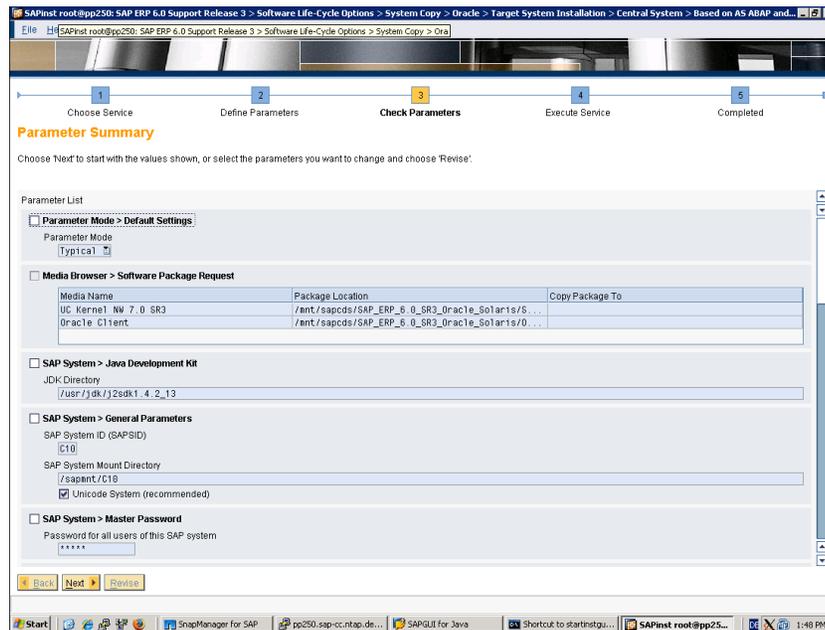


Figure 17) SAPinst—parameter summary.

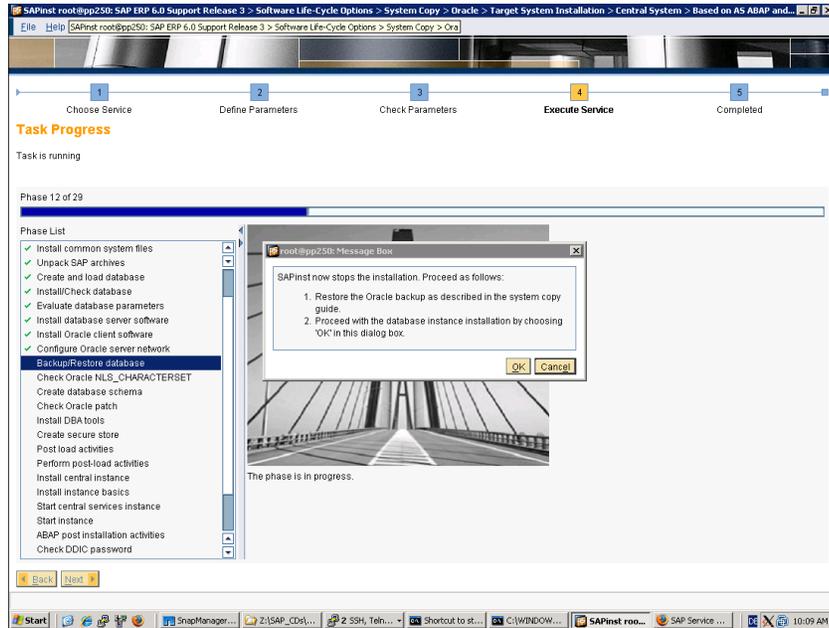


Figure 18) SAPinst—stops to restore the database.

SAPinst stops while SMSAP is used to clone the database.

### CREATE CLONE USING SMSAP

The cloning process is accomplished with the following steps.

- Creation of an online backup of the source system
- Clone wizard needs the following input:
  - Target SID and host
  - Data file mount points
  - Redo log groups and members
  - Init<SID>.ora settings
  - “SAP system copy follow-up activities” as a postcloning step. This script will amongst other things delete the background jobs on the target system. The plug-in examples can be found at `/opt/NetApp/smsap/plugins/examples/clone/create/post` and have to be copied to the plug-in directory `/opt/NetApp/smsap/plugins/clone/create/post`.
  - SCHEMAOWNER information
  - Oracle® user settings
- Cloning specifications are saved for future use

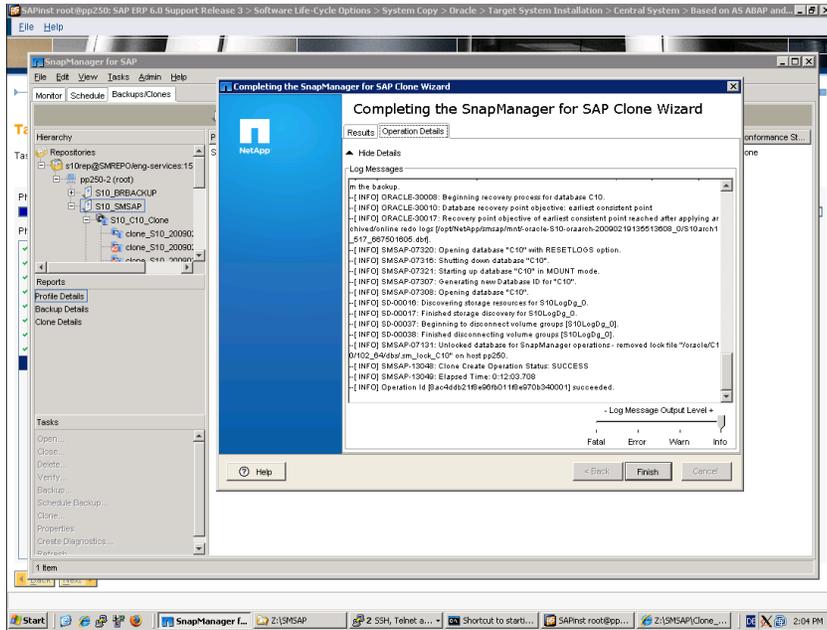


Figure 19) SMSAP—cloning.

The cloning operation is finished in around 12 minutes in the described setup.

## CONTINUE SAPINST ON THE TARGET HOST

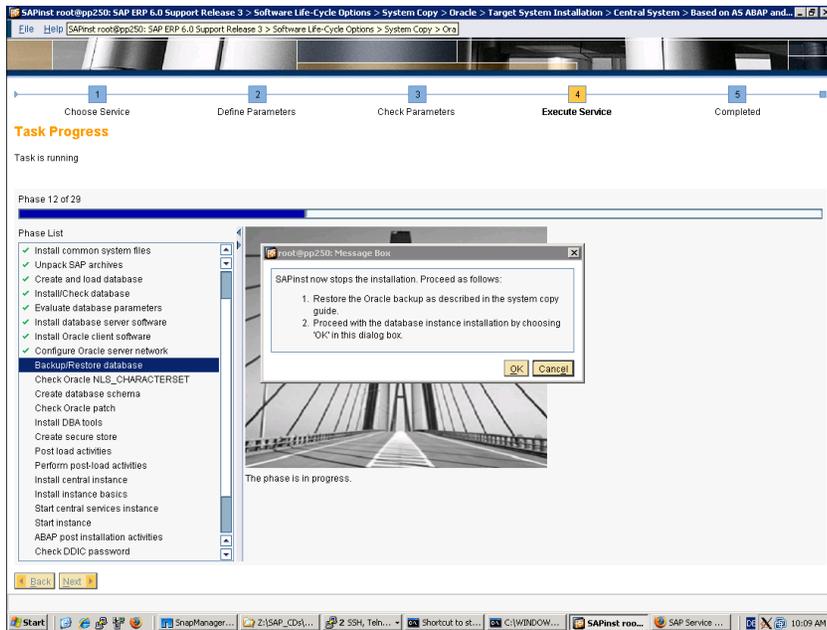


Figure 20) SAPInst—continue installation.

Continue SAPInst.



### 3.3 REFRESH OF TARGET SYSTEM

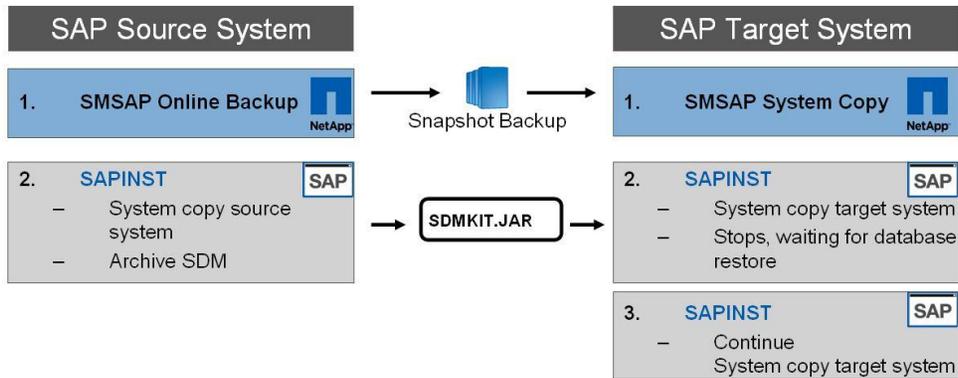


Figure 22) SAP system copy double stack, system refresh, NetApp approach.

#### SAPINST ON THE SOURCE SYSTEM

If SDM data changed, SAPinst needs to run on the source system to archive the current SDM data. SAPinst runs in unattended mode as described in section 3.1. The SDM data will be available in the export location.

#### STOP SAP SYSTEM ON THE TARGET HOST

```
bash-3.00# su - c10adm -c "stopsap r3"
```

Database must **not** be shut down, since SMSAP requires a running database.

#### DELETE CLONE USING SMSAP

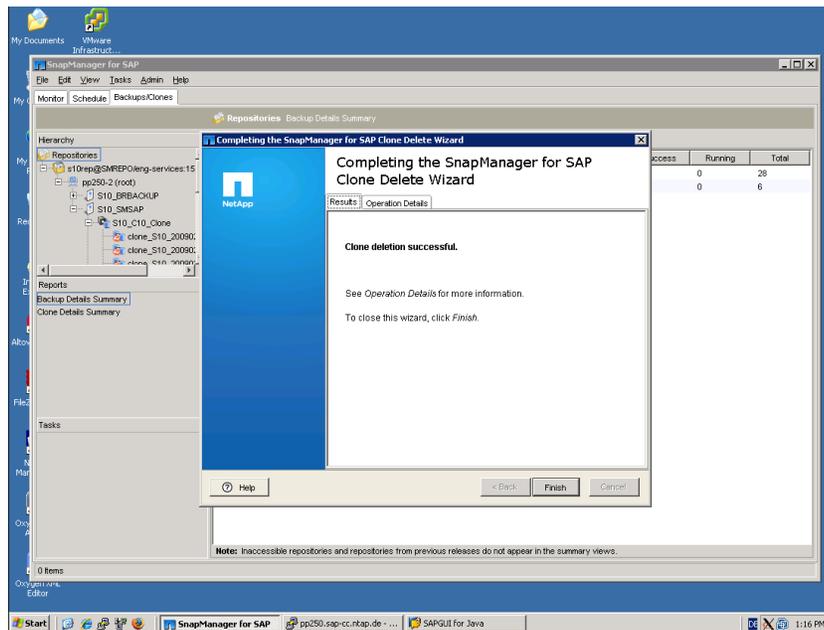


Figure 23) SMSAP—delete clone.

## CREATE CLONE USING SMSAP

The cloning process is accomplished with the following steps.

- Creation of an online backup of the source system
- Clone wizard needs the following input:
  - Target SID and host
  - Clone specification file that was saved with first system copy as described in section 3.2.

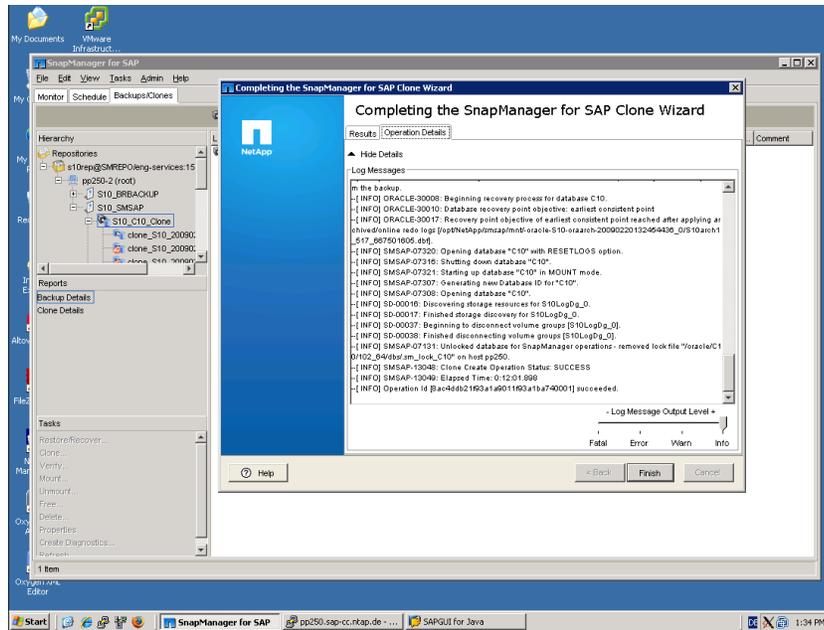


Figure 24) SMSAP—create clone.

The clone operation is finished in around 12 minutes in our setup.

## SAPINST ON THE TARGET HOST

The input for SAPInst was saved during the first run of SAPInst, as described in section 3.2.

The saved data in the template directory will be used to run SAPInst with minimal user input.

```
bash-3.00# mkdir Another_New_SAPinst_Dir
bash-3.00# chmod 777 ./Another_New_SAPinst_Dir
bash-3.00# chown root:SAPinst ./Another_New_SAPinst_Dir
bash-3.00# cp -pr ./golden_template/* ./Another_New_SAPinst_Dir
```

SAPinst is started in the new directory "Another\_New\_SAPinst\_Dir."

```
bash-3.00#
mnt/sapcds/SAP_ERP_6.0_SR3_Oracle_Solaris/Installation_Master/IM_SOLARIS_SPARC/SAPinst
-noGUI
```

SAPinst now skips all the parameter input and the execution of the service can be started immediately.

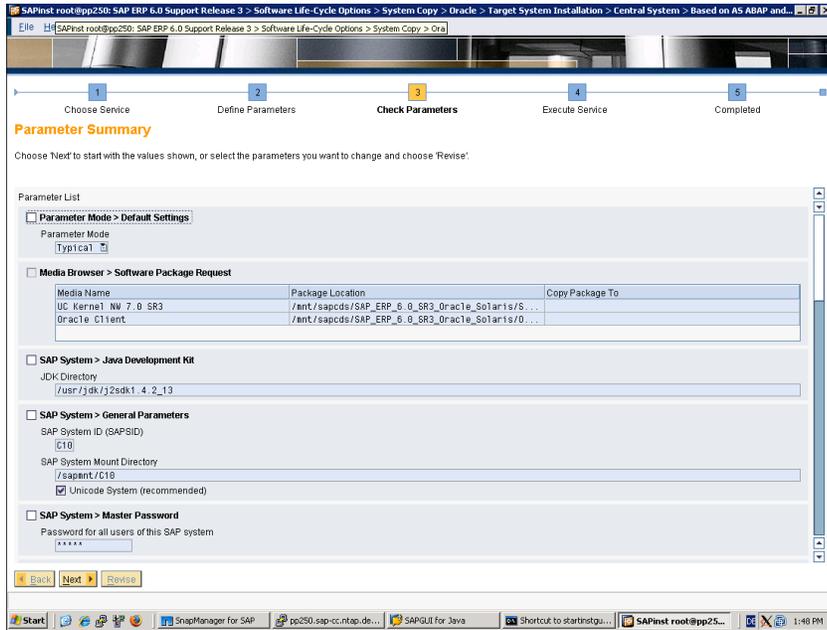


Figure 25) SAPinst—parameter summary.

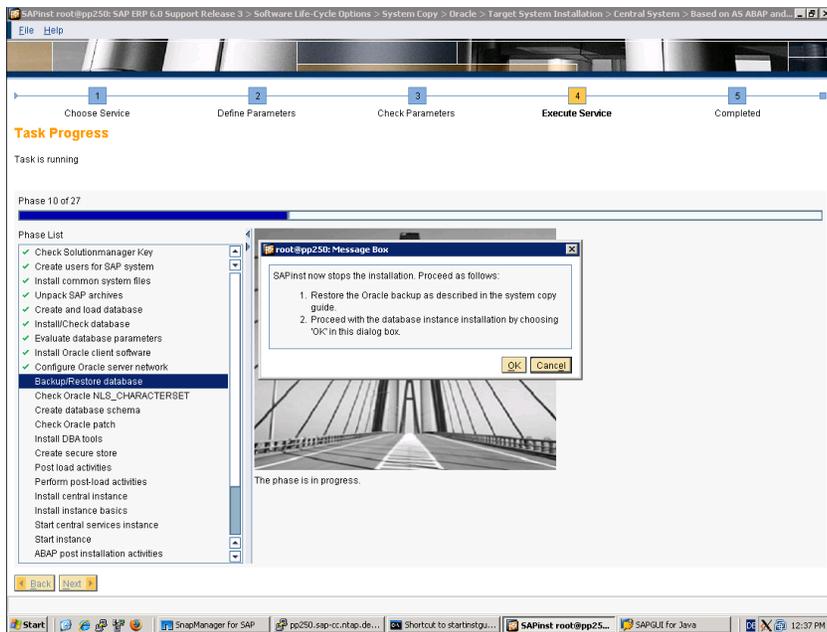


Figure 26) SAPinst—stops to restore the database.

Since the database was already made available by SMSAP, the installation can be continued immediately.

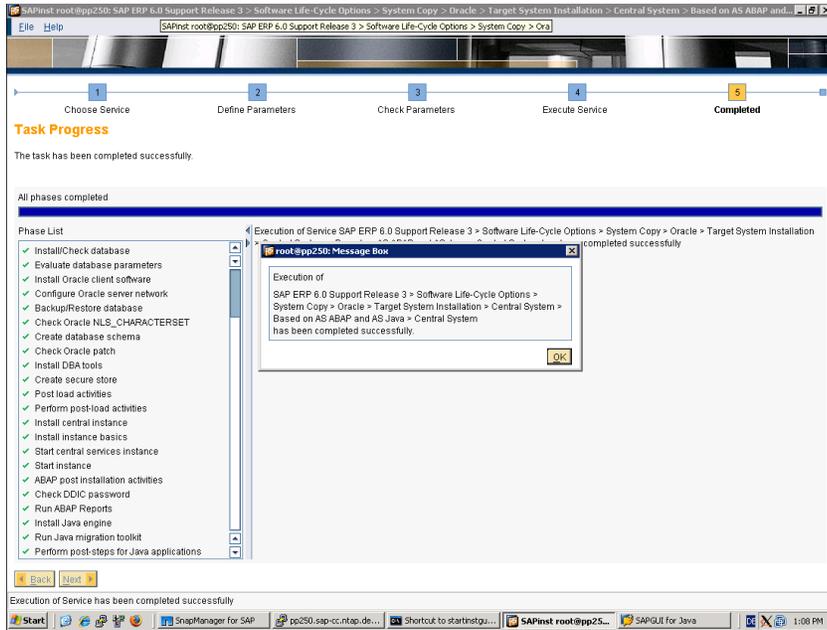


Figure 27) SAPinst—installation completed.

The SAP system copy is finished. Logging on to the new SAP system is possible.

Runtime is around 25 minutes in the described setup.

## 4 SAP SYSTEM COPY ABAP STACK ONLY

### 4.1 TASKS ON THE SOURCE SYSTEM

#### ORABRCOPY TOOL

There is no need to run the OraBrCopy tool on the source system, since SMSAP handles the change of the SID on the target system. A generic CONTROL.SQL is used and needs to be made available when SAPinst is executed on the target host. The file must have the following content.

```
bash-3.00# cat CONTROL.SQL
REM =====
REM CONTROL.SQL
REM =====
STARTUP
```

### 4.2 NEW INSTALLATION OF THE TARGET SYSTEM

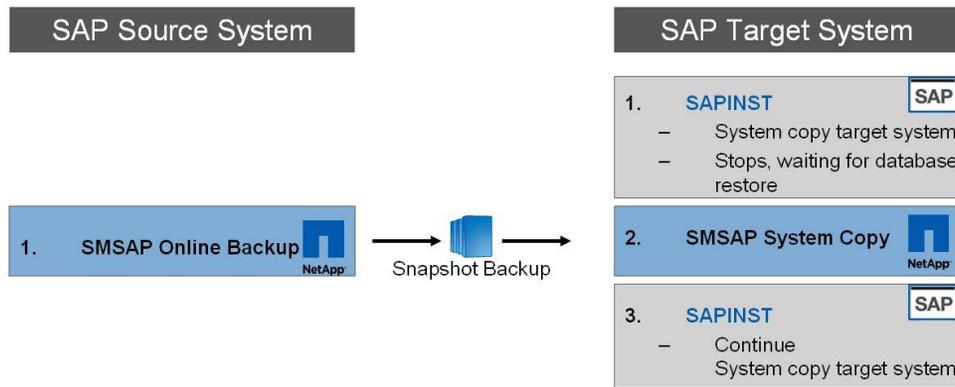


Figure 28) SAP system copy ABAP stack, new installation, NetApp approach.

#### BASIC SETUP

All necessary file systems except the sapdata file systems need to be set up as described in the SAP installation guide. The sapdata file systems will be connected later using a FlexClone copy.

#### SDU Installation

- Configuration access to storage controller(s)

```
bash-3.00# snapdrive config set root sapfiler2
Password for root:
Retype password:
bash-3.00#
```

- Configuration of FlexClone for cloning in snapdrive.conf configuration file

```
san-clone-method="optimal" # Clone methods for snap connect
```

#### SMSAP Installation

- No additional configuration is necessary.

## SAPINST ON TARGET HOST

SAPinst is used to install the SAP system using the homogenous system copy method described in the previous section, “SAP System Copy ABAP and Java Stack.” There is no need to save the input made in SAPinst, since SAPinst is only used for the “New installation” process. When refreshing the target system there is no need to run SAPinst any more.

SAPinst will install the target system and then ask for the restore of the database.

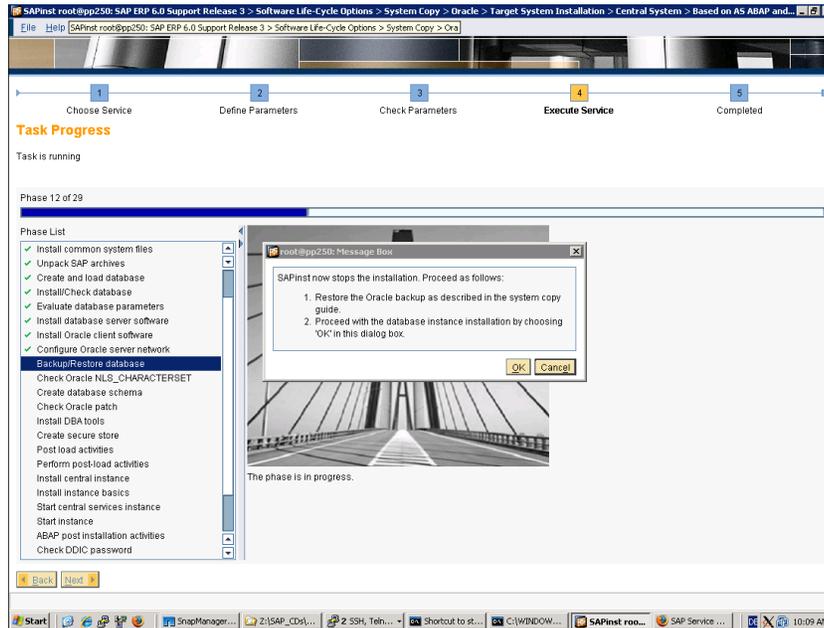


Figure 29) SAPinst—stops to restore the database.

Keep SAPinst running while SMSAP is used to clone the database.

## CREATE CLONE USING SMSAP

The cloning process is accomplished with the following steps.

- Creation of an online backup of the source system
- Clone wizard needs the following input:
  - Target SID and host
  - Data file mount points
  - Redo log groups and members
  - initSID.ora settings
  - “SAP system copy follow-up activities” as a postcloning step. This script will amongst other things delete the background jobs on the target system. The plug-in examples can be found at `/opt/NetApp/smsap/plugins/examples/clone/create/post` and have to be copied to the plug-in directory `/opt/NetApp/smsap/plugins/clone/create/post`.
  - SCHEMAOWNER information
  - Oracle user settings
- The cloning specification is saved for future use

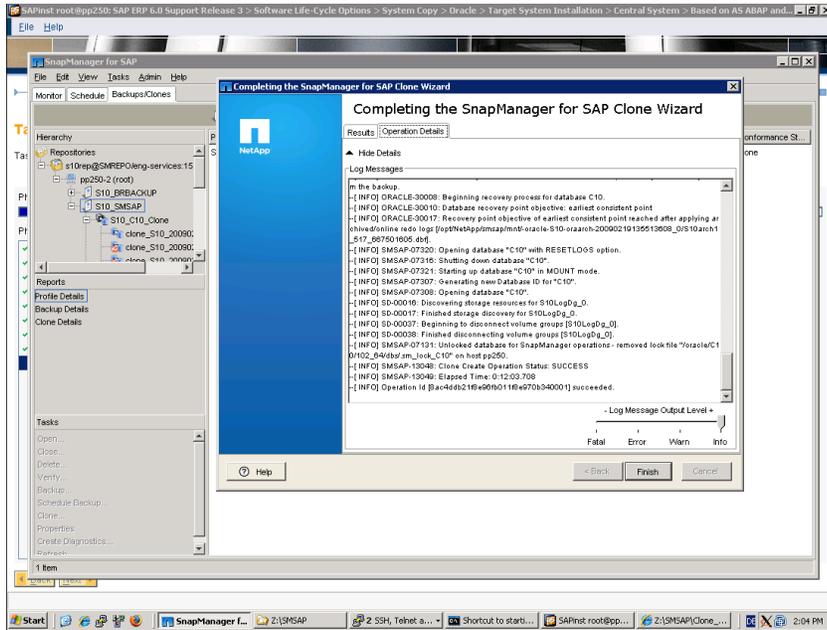


Figure 30) SMSAP—create clone.

The clone operation is finished.

### CONTINUE SAPINST ON THE TARGET HOST

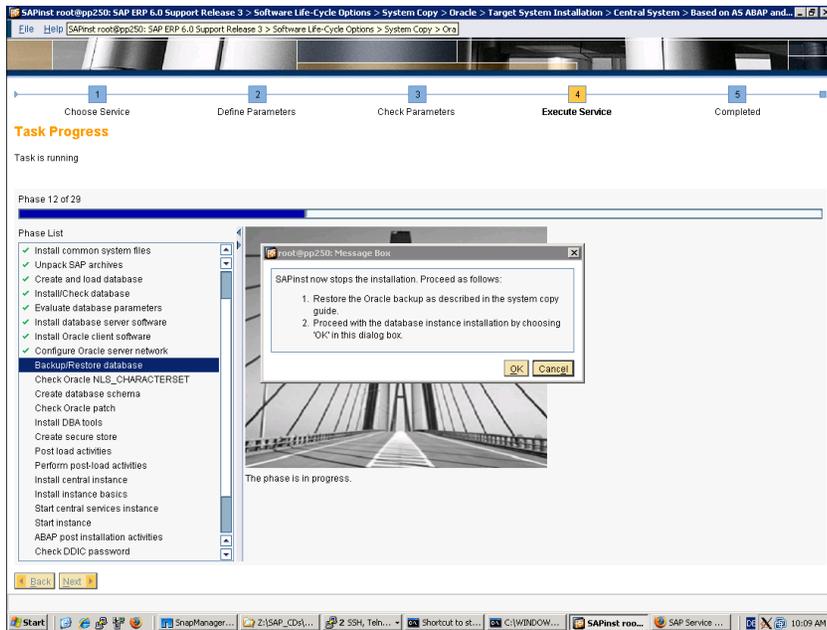


Figure 31) SAPInst—installation completed.

Continue SAPInst.

When SAPInst is finished, the database and the SAP system are running.

The SAP system copy is finished. Logging on to the new SAP system is possible.

### 4.3 REFRESH OF TARGET SYSTEM

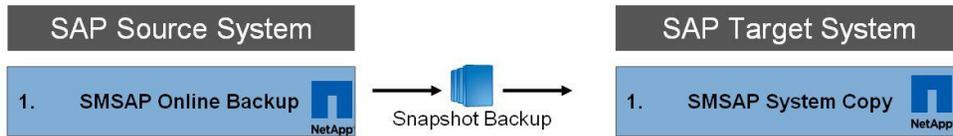


Figure 32) SAP system copy ABAP stack, system refresh, NetApp approach.

#### STOP SAP SYSTEM ON THE TARGET HOST

```
bash-3.00# su - c10adm -c "stopsap r3"
```

Database must **not** be shut down, since SMSAP requires a running database.

#### DELETE CLONE USING SMSAP

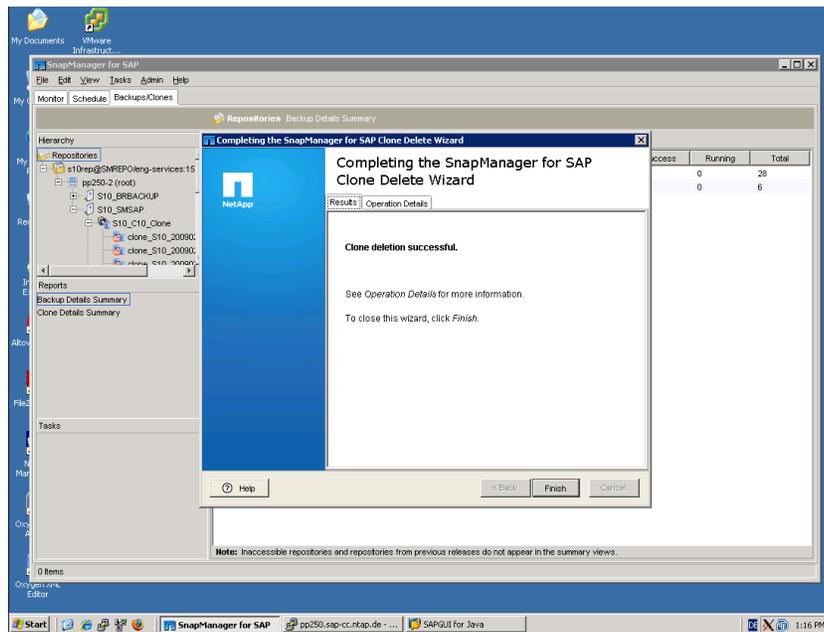


Figure 33) SMSAP—delete clone.

## CREATE CLONE USING SMSAP

The cloning process is accomplished with the following steps.

- Creation of an online backup of the source system
- Clone wizard needs the following input:
  - Target SID and host
  - Clone specification file that was saved with the first system copy as described in section 4.2
  - Since SAPinst is not used when refreshing the target system, the OPS\$ configuration needs to be done by SMSAP. Select “Oracle User for OS-Based DB Authentication” as a postcloning step. This script is used to run the ORADBUSR.SQL, which is provided by SAP.
  - The script ORADBUSR.SQL can be downloaded from the SAP service marketplace. See SAP Notes 361641 and 50088. The script should be copied to the plug-in directory /opt/NetApp/smsap/plugins/clone/create/post.

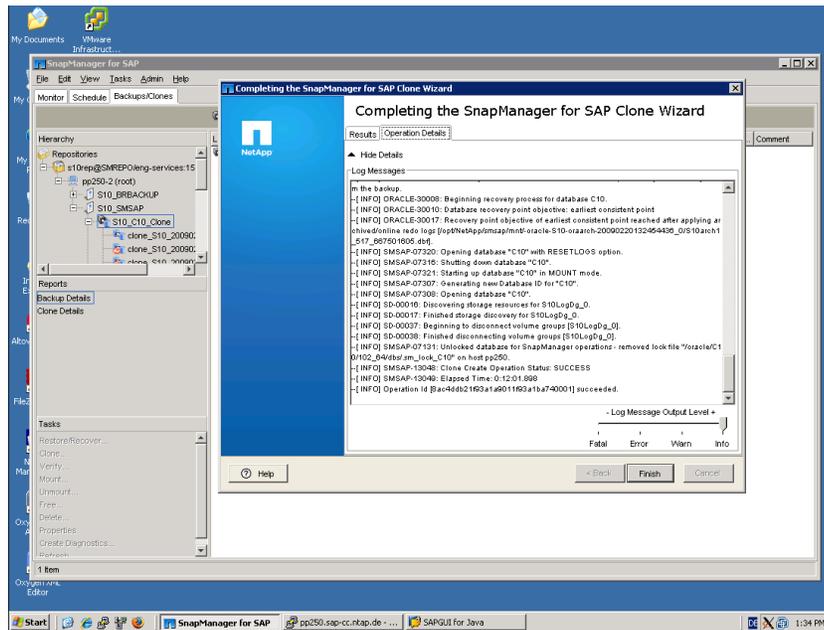


Figure 34) SMSAP—create clone.

The clone operation is finished in around 12 minutes in the described setup.

The SAP system copy is finished. Logging on to the new SAP system is possible.

## 5 SETUP OVERVIEW

The figure below shows the setup that was used for this technical report. The source and target system are configured with FCP storage access.

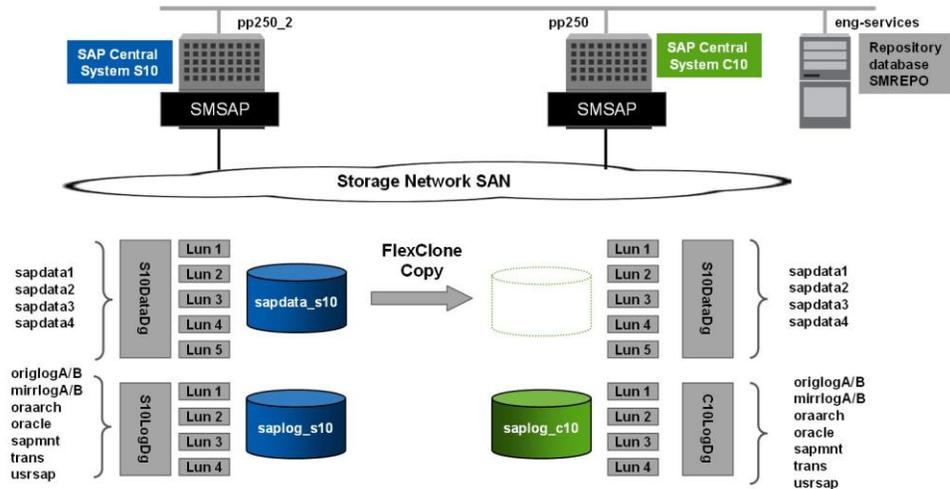


Figure 35) Setup overview.

Both the source and target system are installed using the following software versions:

- Solaris™ 10 update 4
- Veritas™ 5.0 MP RP3
- SAP ERP 6.0 (NW 2004s SR3)
- Oracle 10.2
- Database size: ~100GB
- SMSAP 3.0.2 and SDU 4.1.1

### SMSAP CLONE SPECIFICATION FILE

This sample clone specification file also contains the postcloning step to configure the OPS\$ mechanism. This step is only necessary when refreshing an ABAP-only SAP system. For Java-based systems and for a new installation of an ABAP system, this postcloning step must not be included.

```

clone-specification xmlns="http://www.netapp.com">
  <database-specification>
    <controlfiles>
      <file>/oracle/C10/origlogA/cntrl/cntrlC10.dbf</file>
      <file>/oracle/C10/origlogB/cntrl/cntrlC10.dbf</file>
      <file>/oracle/C10/mirrlogA/cntrl/cntrlC10.dbf</file>
    </controlfiles>
    <redologs>
      <redogroup>
        <file>/oracle/C10/origlogA/log_g11m1.dbf</file>
        <file>/oracle/C10/origlogA/log_g13m1.dbf</file>
        <number>1</number>
        <size unit="M">50</size>
      </redogroup>
      <redogroup>
        <file>/oracle/C10/origlogB/log_g12m1.dbf</file>
        <file>/oracle/C10/origlogB/log_g14m1.dbf</file>
        <number>2</number>
        <size unit="M">50</size>
      </redogroup>
    </redologs>
  </database-specification>

```

```

</redogroup>
<redogroup>
  <file>/oracle/C10/mirrlogA/log_g11m2.dbf</file>
  <file>/oracle/C10/mirrlogA/log_g13m2.dbf</file>
  <number>3</number>
  <size unit="M">50</size>
</redogroup>
<redogroup>
  <file>/oracle/C10/mirrlogB/log_g12m2.dbf</file>
  <file>/oracle/C10/mirrlogB/log_g14m2.dbf</file>
  <number>4</number>
  <size unit="M">50</size>
</redogroup>
</redologs>
<parameters>
  <parameter>
    <name>background_dump_dest</name>
    <value>/oracle/C10/saptrace/background</value>
  </parameter>
  <parameter>
    <name>core_dump_dest</name>
    <value>/oracle/C10/saptrace/background</value>
  </parameter>
  <parameter>
    <name>user_dump_dest</name>
    <value>/oracle/C10/saptrace/usertrace</value>
  </parameter>
  <parameter>
    <name>query_rewrite_enabled</name>
    <value>>false</value>
  </parameter>
  <parameter>
    <name>remote_login_passwordfile</name>
    <value>EXCLUSIVE</value>
  </parameter>
  <parameter>
    <name>log_archive_dest</name>
    <value>/oracle/C10/oraarch/</value>
  </parameter>
  <parameter>
    <name>_in_memory_undo</name>
    <value>FALSE</value>
  </parameter>
  <parameter>
    <name>control_file_record_keep_time</name>
    <value>30</value>
  </parameter>
  <parameter>
    <name>log_buffer</name>
    <value>14346240</value>
  </parameter>
  <parameter>
    <name>processes</name>
    <value>80</value>
  </parameter>
  <parameter>
    <name>sort_area_retained_size</name>
    <value>0</value>
  </parameter>
  <parameter>
    <name>_index_join_enabled</name>
    <value>FALSE</value>
  </parameter>
  <parameter>
    <name>db_files</name>
    <value>254</value>
  </parameter>
  <parameter>
    <name>_sort_elimination_cost_ratio</name>
    <value>10</value>
  </parameter>
  <parameter>
    <name>undo_tablespace</name>
    <value>PSAPUNDO</value>
  </parameter>
  <parameter>
    <name>log_checkpoints_to_alert</name>
    <value>TRUE</value>
  </parameter>

```

```

</parameter>
<parameter>
  <name>sort_area_size</name>
  <value>2097152</value>
</parameter>
<parameter>
  <name>_optim_peek_user_binds</name>
  <value>FALSE</value>
</parameter>
<parameter>
  <name>recyclebin</name>
  <value>off</value>
</parameter>
<parameter>
  <name>star_transformation_enabled</name>
  <value>>true</value>
</parameter>
<parameter>
  <name>sga_max_size</name>
  <value>1224736768</value>
</parameter>
<parameter>
  <name>db_cache_size</name>
  <value>587202560</value>
</parameter>
<parameter>
  <name>pga_aggregate_target</name>
  <value>773094113</value>
</parameter>
<parameter>
  <name>undo_retention</name>
  <value>43200</value>
</parameter>
<parameter>
  <name>shared_pool_size</name>
  <value>587202560</value>
</parameter>
<parameter>
  <name>parallel_execution_message_size</name>
  <value>16384</value>
</parameter>
<parameter>
  <name>_table_lookup_prefetch_size</name>
  <value>0</value>
</parameter>
<parameter>
  <name>shared_pool_reserved_size</name>
  <value>57982058</value>
</parameter>
<parameter>
  <name>open_cursors</name>
  <value>800</value>
</parameter>
<parameter>
  <name>log_checkpoint_interval</name>
  <value>0</value>
</parameter>
<parameter>
  <name>sessions</name>
  <value>96</value>
</parameter>
<parameter>
  <name>_b_tree_bitmap_plans</name>
  <value>FALSE</value>
</parameter>
<parameter>
  <name>undo_management</name>
  <value>AUTO</value>
</parameter>
<parameter>
  <name>replication_dependency_tracking</name>
  <value>FALSE</value>
</parameter>
<parameter>
  <name>event</name>
  <value>10191 trace name context forever, level 1</value>
</parameter>
<parameter>
  <name>filesystemio_options</name>
  <value>setall</value>

```

```

</parameter>
<parameter>
  <name>dml_locks</name>
  <value>4000</value>
</parameter>
<parameter>
  <name>remote_os_authent</name>
  <value>TRUE</value>
</parameter>
<parameter>
  <name>job_queue_processes</name>
  <value>1</value>
</parameter>
<parameter>
  <name>_optimizer_mjc_enabled</name>
  <value>FALSE</value>
</parameter>
</parameters>
<oracle-home>/oracle/C10/102_64</oracle-home>
<oracle-os-account>
  <username>oracl0</username>
  <group>dba</group>
</oracle-os-account>
<sql-statements/>
</database-specification>
<storage-specification>
<storage-mapping>
  <mountpoint>
    <source>/oracle/S10/sapdata4</source>
    <destination>/oracle/C10/sapdata4</destination>
  </mountpoint>
  <mountpoint>
    <source>/oracle/S10/sapdata3</source>
    <destination>/oracle/C10/sapdata3</destination>
  </mountpoint>
  <mountpoint>
    <source>/oracle/S10/sapdata2</source>
    <destination>/oracle/C10/sapdata2</destination>
  </mountpoint>
  <mountpoint>
    <source>/oracle/S10/sapdata1</source>
    <destination>/oracle/C10/sapdata1</destination>
  </mountpoint>
</storage-mapping>
</storage-specification>
<task-specification>
<post-tasks>
  <task>
    <name>SAP SystemCopy follow-up activities</name>
    <parameter>
      <name>SCHEMAOWNER</name>
      <value>SAPSR3</value>
    </parameter>
  </task>
  <task>
    <name>Oracle Users for OS based DB authentication</name>
    <parameter>
      <name>SCHEMAOWNER</name>
      <value>SAPSR3</value>
    </parameter>
    <parameter>
      <name>ORADBUSR_FILE</name>
      <value>/opt/NetApp/smsap/plugins/clone/create/post/ORADBUSR.SQL</value>
    </parameter>
  </task>
</post-tasks>
</task-specification>
</clone-specification>

```

The postcloning task to configure the OPS\$ mechanism is only needed when refreshing an ABAP-only system. For Java-based systems and for a new installation of an ABAP system, this postcloning step must not be included.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and

© Copyright 2010 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, FlexClone, SnapDrive, SnapManager, SnapMirror, Snapshot, and SnapVault are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Java and Solaris are trademarks of Sun Microsystems, Inc. SAP is a registered trademark of SAP AG. SQL is a registered trademark of Microsoft Corporation. Veritas is a trademark of Symantec Corporation. Oracle is a registered trademark of Oracle Corporation. All other brands or products are trademark or registered trademarks of their respective holders and should be treated as such. TR-3812



[www.netapp.com](http://www.netapp.com)