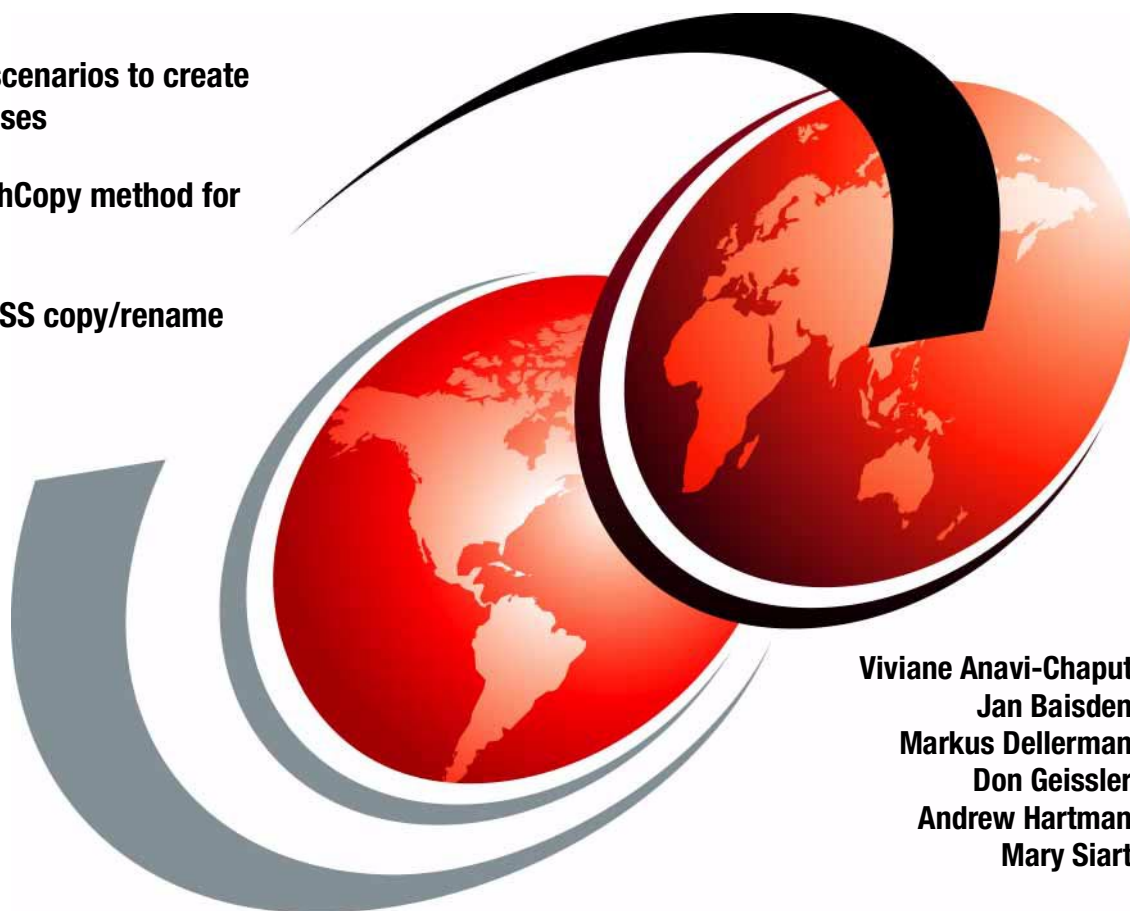


SAP on DB2 for z/OS and OS/390: DB2 System Cloning

Hands-on scenarios to create
test databases

Online FlashCopy method for
ESS disks

Offline DFDSS copy/rename
method



Viviane Anavi-Chaput
Jan Baisden
Markus Dellerman
Don Geissler
Andrew Hartman
Mary Siart



International Technical Support Organization

**SAP on DB2 for z/OS and OS/390: DB2 System
Cloning**

November 2001

Take Note! Before using this information and the product it supports, be sure to read the general information in “Special notices” on page 123.

First Edition (November 2001)

This edition applies to SAP R/3 V4.6C for use with DB2 for z/OS and OS/390 V6.1 operating on the OS/390 V2R10 platform.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2001. All rights reserved.

Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	vii
The team that wrote this redbook	vii
Special notice	ix
IBM trademarks	x
Comments welcome	x
Chapter 1. Introduction to SAP homogeneous system copy	1
1.1 Types of SAP R/3 system copy	2
1.1.1 Heterogeneous system copy	2
1.1.2 Homogeneous system copy	2
1.2 What triggers the need for HSC?	3
1.2.1 Application testing and quality assurance	3
1.2.2 System function testing	4
1.2.3 Production maintenance	4
1.2.4 Reporting	5
1.2.5 Data mining	5
1.2.6 Training	5
1.3 HSC methodologies	6
1.3.1 Online copy methodology	6
1.3.2 Offline copy methodology	7
1.3.3 Choosing a method	8
1.4 Introducing the HSC test scenarios for SAP	10
1.4.1 Scenario 1 - FlashCopy with suspended DB2	10
1.4.2 Scenario 2 - DFSS copy with offline DB2	12
1.4.3 Implementation steps for HSC process	13
Chapter 2. Planning for DB2 system cloning	15
2.1 Planning for online copy	16
2.1.1 Timing of copying the data	16
2.1.2 UCAT and SMS storage group configurations	16
2.1.3 Skeleton BSDS for the target system	16
2.1.4 Online copy steps	19
2.2 Planning for offline copy	20
Chapter 3. Designing the DB2 system cloning process	23
3.1 Configuration considerations	24
3.1.1 Data sharing to data sharing	24
3.1.2 Data sharing to non-data sharing	26
3.1.3 Non-data sharing to non-data sharing	27

3.1.4 Non-data sharing to data sharing	27
3.2 DB2 system cloning phases	27
3.2.1 Checking the source environment.	27
3.2.2 Copying the source system.	28
3.2.3 Preparing the target disk environment	29
3.2.4 Restoring and building target disks.	29
3.2.5 Restarting the target DB2 system	29
3.2.6 Restarting the target SAP system.	30
3.2.7 Verifications and backup	30
Chapter 4. Scenario 1 - FlashCopy with suspended DB2	33
4.1 Homogeneous system copy using FlashCopy.	34
4.2 Checking the source environment	35
4.3 Copying the source system	36
4.4 Preparing the target disk environment	38
4.5 Restoring and building target disks.	41
4.6 Restarting the target DB2 system.	42
4.6.1 Collecting restart information for the target DB2 system.	43
4.6.2 Restarting the target DB2 system	44
4.6.3 Installation customizations	51
4.7 Restarting target SAP system.	52
4.8 Verifications and backup	53
Chapter 5. Scenario 2 - DFDSS copy with offline DB2.	55
5.1 Homogeneous system copy using DFDSS copy	56
5.2 Preparing the target disk environment	57
5.3 Checking the source environment	58
5.4 Copying the source system	59
5.5 Restarting target DB2 system.	59
5.6 Restarting the target SAP system.	62
5.7 Verifications and backup	63
Appendix A. JCL for Scenario 1 - FlashCopy with suspended DB2.	65
Appendix B. JCL for Scenario 2 - DFDSS copy with offline DB2	93
Appendix C. REXX procedures	105
Appendix D. Additional material	119
Locating the Web material	119
Using the Web material	119
Related publications	121
IBM Redbooks	121
Other resources	121

Referenced Web sites	121
How to get IBM Redbooks	121
IBM Redbooks collections	122
Special notices	123
Abbreviations and acronyms	125
Index	127

Preface

This IBM Redbook shows the different techniques customers can use to clone their DB2 system. Although the scenarios in the book use an SAP workload, the DB2 cloning techniques we describe are not specific to SAP. Those techniques can apply to any DB2 system and the applications running on that DB2. The book describes hands-on details of selected DB2 database cloning scenarios, typically the most popular techniques currently used by customers, as well as the most efficient scenarios we would like to recommend to DB2 database administrators.

The scenarios described in this book are based on SAP examples. Our source system resides on a DB2 data sharing group. We have one target system residing on another DB2 data sharing group and one residing on a non-data sharing DB2. All source and target systems are in the same OS/390 Sysplex environment.

We discuss and show DB2 system cloning techniques using the following methods:

- ▶ ESS FlashCopy
- ▶ DFSMS copy and rename

The book addresses an audience with in depth knowledge of DB2 for z/OS and OS/390 database administration and familiarity with SAP system requirements.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Viviane Anavi-Chaput is a Senior IT Specialist for BI, ERP and DB2 at the IBM International Technical Support Organization, Poughkeepsie Center. She writes extensively, teaches worldwide, and presents at international conferences on all areas of Business Intelligence and DB2 for OS/390. Before joining the ITSO in 1999, Viviane was a Senior Data Management Consultant at IBM Europe, France. She was also an ITSO Specialist for DB2 at the San Jose Center from 1990 to 1994.

Jan Baisden is a Senior Market Support Representative at the International Technical Support Organization, Poughkeepsie Center. He concentrates on ERP solutions and particularly on AIX support of S/390 usages in those solutions. He currently maintains five redbooks on SAP R/3 on DB2 for OS/390. Before joining the ITSO six years ago, Jan worked at the IBM International Systems Center in Gaithersburg, Maryland, supporting projects in Latin America, Asia, and Australia.

Markus Dellermann is a DB2 systems programmer at INA Werk Schaeffler oHG in Herzogenaurach, Germany. He has 3 years of experience with DB2 including 1 1/2 years with SAP R/3 on DB2 for OS/390. He holds a degree in economics and computer science.

Don Geissler is an IBM Consulting Technical Sales Specialist in ERP Solutions Sales of North America. He has been with IBM since 1983, providing systems engineering support to large S/390 customers. Since September 1996 Don has been a member of the ERP Competency Center S/390 team, working with SAP R/3 for DB2 for OS/390. His primary responsibilities are consulting with SAP R/3 for OS/390 customers in the areas of high availability, database backup and recovery, and disaster recovery.

Andrew Hartman is a DB2 Systems Programmer working for the Standard Bank Of South Africa in Johannesburg, South Africa. He has 21 years of experience in the I/T industry specializing in various databases. He has worked extensively on DB2 for OS/390 for the past 11 years and SAP on DB2 for OS/390 for 1 1/2 years.

Mary Siart is an Advisor Systems Programmer for Motorola in the USA. She has 22 years of extensive experience in the I/T industry, specializing mainly in the data management areas. Before joining Motorola, Mary worked for IBM as a systems engineer supporting IMS and MVS. At Motorola, Mary is working with SAP on OS390 since 1996. Her primary focus has been to develop an SAP infrastructure for OS390 which provides high availability and continuous operation.

Thanks to the following people for their contributions to this project:

Rich Conway
IBM International Technical Support Organization, Poughkeepsie Center

Mike Gordon
SAP for OS/390 - DB2 Architect, US

Namik Hrle
SAP on IBM zSeries Software Development, Boeblingen, Germany

Susan W Lynch
SAP system test, IBM US

Andreas Maier
SAP on IBM zSeries Software Development, Boeblingen, Germany

Manfred Olschanowsky
IBM/SAP Integration, IBM Silicon Valley Laboratory, US

Andrea Reidel
SAP Solution Test, IBM Germany

Helmut Schilling
IBM z/Series SW Test-Lab, Boeblingen, Germany

James Teng
DB2 development, IBM Silicon Valley Laboratory, US

Emil Wolf
IBM z/Series SW Test-Lab, Boeblingen, Germany

Michael Zazulak
IBM/SAP Competency Center, US

Jochen Ziegler
IBM Certified IT Expert for SAP, Germany

Thanks also to Al Schwab and Terry Barthel for their editorial assistance, and to Ella Buslovich for her graphics assistance.

Special notice

This publication is intended to help SAP database administrators to do homogeneous system copies of their SAP system. The information in this publication is not intended as the specification of any programming interfaces that are provided by SAP R/3 or DB2 for OS/390 products. See the PUBLICATIONS section of the SAP and IBM Programming Announcements for more information about what publications are considered to be product documentation.

IBM trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

e (logo)® 

IBM ®

AIX®

Database 2™

DB2®

DFS™

Enterprise Storage Server™

ESCON®

FICON™


FlashCopy™

IMS™

MVS™

OS/390®

Redbooks™

Redbooks Logo 

Perform™

RACF®

RAMAC®

RS/6000®

S/390®

SP™

System/390®

z/OS™

zSeries™

Lotus®

Comments welcome

Your comments are important to us!

We want our IBM Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:

ibm.com/redbooks

- ▶ Send your comments in an Internet note to:

redbook@us.ibm.com

- ▶ Mail your comments to the address on page ii.



Introduction to SAP homogeneous system copy

This chapter introduces SAP homogeneous system copy. We discuss why customers need to copy their SAP R/3 systems. We define the different types of SAP R/3 system copies, expanding more specifically on the SAP R/3 homogeneous system copy. The chapter includes the following introductory sections:

- ▶ SAP R/3 system copy types
- ▶ What triggers the need for HSC?
- ▶ HSC methodologies
- ▶ Introducing the HSC test scenarios

1.1 Types of SAP R/3 system copy

SAP R/3 supports two types of system copy:

- ▶ Heterogeneous system copy
- ▶ Homogeneous system copy

1.1.1 Heterogeneous system copy

A heterogeneous system copy is used for making a copy of an SAP R/3 system and changing the operating system and/or the database server. This type of system copy is generally referred to as a system *migration*. We do not describe heterogeneous system copy because it is outside the scope of this redbook.

1.1.2 Homogeneous system copy

HSC is used for making a copy of an SAP R/3 system where the operating system and the database server are the same on the target system. HSC is sometimes referred to as a system *clone*.

In HSC, the target SAP R/3 database server has the same operating system (such as OS/390 or zOS) and the same database system (such as DB2 for OS/390) as the source SAP R/3 system. The contents of the database are copied from the source SAP R/3 system to the target SAP R/3 system. The SAP and DB2 version/release of the source and target systems must also be the same.

SAP R/3 supports two methods for performing an HSC:

- ▶ Using SAP R/3 Export/Import tools
- ▶ Using database-specific tools

The SAP R/3 Export/Import procedure uses a standard SAP-supplied transaction to export the data from the source database to a flat file, and then to import the data into the target database. This process is not recommended for large production SAP environments. Typically, this process is used with small systems that are being used in pilot projects or some development efforts. The time to accomplish the export-import process with large production systems makes this process prohibitive. We do not cover the SAP R/3 Export/Import procedure in this redbook.

1.2 What triggers the need for HSC?

Customers use SAP R/3 homogeneous system copy for the following several reasons:

- ▶ Application testing and quality assurance
- ▶ System function test
- ▶ Production maintenance
- ▶ Reporting
- ▶ Data mining
- ▶ Training

1.2.1 Application testing and quality assurance

This is the major reason for doing an HSC.

Some SAP R/3 installations may have several application teams in some stage of development and testing. Each team has the potential to corrupt the database, or to interfere with the data being used by other teams. Thus each team may require its own test copy of the system.

These *application test* system copies may have a variety of purposes, one example being to provide an SAP R/3 system for teams doing the next phase of application development, where there is a possibility that a team inadvertently removes the SAP R/3 database.

When you are planning to add additional workload on your system, we recommend that you perform a load test to measure the impact on the existing system. SAP R/3 homogeneous system copy can be used to create this load test environment. The life expectancy of this type of system copy would depend on the extent of load testing that needs to be done.

When you are taking a phased approach to implementing SAP applications in your environment, there could be a need to use system copy to create multiple copies of a development or quality assurance system. Developers and testers could then start with a common base SAP system on which to do development. As each phase of the application implementation cycle ends, a system copy could be used to establish a new base for the next phase of implementation. The life expectancy of this type of system copy would be much longer, usually months.

The SAP R/3 infrastructure (development, quality assurance, and production) can be built as follows:

- ▶ Without using system copy to build the initial production system in the SAP R/3 infrastructure by:

- Installing all SAP R/3 systems and customizing the development system according to the implementation documentation.
- Transporting client-dependent and client-independent data to the quality assurance and production systems.
- ▶ Using system copy to build the initial production system in the SAP R/3 infrastructure by:
 - Installing the development and quality assurance systems according to the implementation documentation.
 - Transporting client-dependent and client-independent data from the development system to the quality assurance system.
 - Installing a central instance for the production system and using SAP R/3 homogeneous system copy to copy the quality assurance system database to the production system database.

The advantage of using HSC is that it ensures that the initial production system is an exact copy of the quality assurance system. From this point on, the transport system is used to copy system configuration changes and user data to the production system.

1.2.2 System function testing

SAP R/3 system copy will also usually be required for testing new releases, simulating disaster recovery, and investigating component loss implications on system availability. These *system-function test* system copies have different characteristics from those used for applications tests.

SAP recommends that when you are planning to upgrade your SAP R/3 system to a new release, you should test the new functions in a test system before doing the upgrade in the existing landscape. SAP R/3 homogeneous system copy provides a way for creating this test environment. The life expectancy of this type of system copy could be several weeks, depending on how many problems are experienced during the upgrade testing.

1.2.3 Production maintenance

You might want to have a copy of your production system to test maintenance before introducing it in the production system. The reason a copy of production is needed is because in most cases the development and quality assurance systems contain configuration and user data differences due to ongoing application development. The only way to truly test a fix for a problem in the production system is to use an exact copy of the production system created by using SAP R/3 homogeneous system copy. The life expectancy of this type of

system copy would be very short, usually only while fix testing is being done. Once the fix has been moved into the production system, the production maintenance system would be of no further use. A new production maintenance system would have to be created the next time a production fix needs testing.

1.2.4 Reporting

Another reason you might want to build a copy of your production system is for heavy month-end or year-end reporting. One advantage for doing this is the reduced activity on the production system. Another advantage is that the copy is unchanged by daily activity. Therefore, reports could be run over several days for the same time period. The life expectancy of this type of system copy would be very short, usually no more than a few days. A new reporting system would be created after month-end or year-end processing has completed.

1.2.5 Data mining

You might also build a copy of your production SAP R/3 system to be used for data warehouse or data mining applications, since production updates could change the underlying data of long-running queries. This also provides the reduction in activity against the production database as discussed in 1.2.4, "Reporting". The life expectancy of this type of system copy would be slightly longer than the reporting system discussed above, probably not more than a week.

1.2.6 Training

The implementation of an SAP R/3 system is a major undertaking for a company. Considerable end user training is required. Several groups of users throughout the organization have to be trained. If several SAP R/3 modules, such as financial, sales and distribution, are going to be implemented in phases over a long period of time, then there will be a long term need for a training system.

Some customers have attempted to use a training client in the quality assurance system to train end users. However, because of configuration changes needed in some cases by the next phase of the implementation, the training client becomes unusable. For this reason, customers have decided to create a dedicated training system.

As each phase of the implementation is completed and tested in the quality assurance system, a training system is created using SAP R/3 homogeneous system copy to make a copy of the quality assurance system.

1.3 HSC methodologies

The method used for HSC is determined by your operational business requirements for both the source system and the target system:

- ▶ If high availability is required for the source system, it will be necessary to get the copy without stopping the SAP application and impacting its availability to the end-users. This method is known as an *online copy*.
- ▶ If high availability is not an issue for the source system, you might be able to afford stopping the source system prior to getting the copy. This means the end-users have found a time when the source SAP application can be unavailable in order to obtain a copy. This method is known as an *offline copy*.

The type of HSC (online or offline) determines the restart method used in the target system. The restart of the target system is either a *conditional restart* or a *cold start*:

- ▶ If the source copy method is an *online copy*, the target restart method is a *conditional restart*.
- ▶ If the source copy method is an *offline copy*, the target restart method is a *cold start*.

Once the source copy and target restart methodologies are determined, it is necessary to design and develop the processes that will be used to accomplish the copy and restart. Many factors go into designing the copy and restart processes, such as:

- ▶ SAP application availability requirements (influence system configurations)
- ▶ Hardware/software resource availability

1.3.1 Online copy methodology

The definition of an online copy implies that the active logs for the source DB2 subsystems are suspended. The SAP end users continue having access to the SAP R/3 applications; however, all update activity is suspended. Update activity is *not* failed and rolled back, but instead looks as though the system is hung for the duration of the suspend. End users performing query-type work do not notice the suspend.

The source DB2 subsystems are suspended by using the DB2 SET LOG SUSPEND command. If the source system is in a data sharing group, the log suspend *must* be obtained across all data sharing group members. The copy is then obtained using hardware-assisted volume-level copies. The copy includes all DB2 directory and catalog objects as well as the ICF catalogs. It may also

include the DB2 system libraries, depending on your installation's configuration. In other words, does each DB2 subsystem have its own copy of the DB2 system libraries, or is there one production copy representing each maintenance level? All disks containing SAP DB2 objects must be copied. Your copy may also include disks containing libraries or DB2 objects for IBM or OEM tools.

The disks with hardware-assisted volume copy allow for nondisruptive availability of the SAP application by providing an inflight copy. An end user performing update types of activities will notice the contention created by the suspend on the active log. The system will appear hung. That is why it is critical to obtain the copy as quickly as possible and resume update activity in the source DB2 subsystems.

The following types of disks have hardware-assisted volume-level copies:

- ▶ ESS FlashCopy
- ▶ RVA Snapshot
- ▶ EMC Timefinder

Using an *online copy* to restore a target system requires us to perform a *conditional restart*. In order to facilitate all phases of a DB2 system restart, it is necessary to register information from the source system in the target system BSDSs. This includes making some of the source system's log records available to the target system, which is accomplished by providing a disk copy of the appropriate source archive log to the target system and registering it via the change log inventory utility. The change log inventory utility is also used in providing a DB2 system checkpoint and the conditional restart point, derived from the source DB2 system, to the target system. With this information from the source DB2 system, the restart of the target DB2 system can resolve incomplete units-of-recovery and pages-to-be-written, thereby providing data consistency.

However, if there was an SAP business process (batch job) running at the time of the source system log suspend, DB2 will roll back only the uncommitted work. DB2 will not roll back the committed work of the SAP business process. In this case, the copy in the target system will have data consistency from a DB2 perspective, but some additional manual actions might be needed in the SAP system, such as cleaning protocol tables and checking the status of batch jobs that were active at the time of log suspend, to restore consistency at the SAP business process level.

1.3.2 Offline copy methodology

The definition of an offline copy implies the source SAP application and DB2 subsystems have been quiesced and stopped. The SAP users do not have access to the source SAP R/3 applications during the copy process.

Once the source SAP application and DB2 database server have been stopped, a copy utility, such as ESS FlashCopy or DFDSS copy/rename, can be used to take copies of all required volumes. This includes all DB2 directory and catalog objects. It may also include the DB2 system libraries, depending on your installation's configuration. In other words, does each DB2 subsystem have its own copy of the DB2 system libraries, or is there one production copy representing each maintenance level?

All disks containing SAP DB2 objects must also be copied. Your copy may also include disks containing libraries or DB2 objects for IBM or OEM tools.

Depending on how long the source system can be down and what media resources are available, the copies will go to either tape or disks. The manner in which the copy is made to disks is dependent on what type of disks and utilities are available to accomplish it.

Typically, in an offline copy for non-ESS disks, DFDSS is used to copy the source system production disks. If copying the source production disk to disk is not feasible, the source system production disks can be copied to tape using DFDSS. The advantage of using the offline copy method is that the restart of the target system is simplified by using a *cold start* instead of a conditional restart.

The *cold start* is a simplified DB2 system restart because we know the data in the source system is consistent at the time of the copy. DB2 does not need to perform forward and backward log recovery. This reduces the amount of information we need to register in the target system's BSDSs via the change log inventory utility (DSNJU003).

With this type of restart, the target system will have data consistency and represent the state of the business at the point in time the SAP application was stopped and the source system copied.

SAP describes the offline HSC method for DB2 for OS/390 in Chapter 4 of the *SAP R/3 Homogeneous System Copy, Release 4.6C SR2* documentation. The procedure described by SAP in this chapter assumes the source and target DB2 system data sets are not SMS-managed and DB2 data sharing is not enabled.

1.3.3 Choosing a method

In your environment, the descriptions, advantages, and characteristics discussed above should lead you to the decision on which method you will use.

It is important for you to achieve consensus within the enterprise on the parameters that will influence the method you will choose. You should build a document that contains considerations that have been agreed upon. As your installation grows and changes, the document should be reviewed. The evolution of your installation may cause you to change some cloning details.

Planners and administrators must be aware of your assumptions regarding:

- ▶ Considerations on the source system and database
- ▶ Considerations on the target system and database

Considerations on the source system and database

You need to consider the following on the source system:

- ▶ Ability to have a consistent database (the possibility to quiesce the database)
- ▶ Acceptable production database outage to produce a clone
- ▶ Batch windows with lightest production use and best resource availability to produce a clone
- ▶ Time when the clone must be available

Once the copy method has been determined, it is necessary to develop the steps for the copy process. The source system configuration will influence the copy process. The process needs to take the following into consideration:

- ▶ The DB2 subsystem type; data sharing versus non-data sharing
- ▶ Type of disks used for the DB2 subsystem objects and system data sets
- ▶ Type of disks used for the SAP DB2 objects
- ▶ Amount of data to be copied
- ▶ Amount of available disks onto which a copy can be made
- ▶ Length of time the copy can be maintained on copy disks
- ▶ Production business processes (jobs) and their schedules

Considerations on the target system and database

You need to consider the following when configuring the target system:

- ▶ Availability requirements for the target system
- ▶ Configuration decisions
- ▶ Are both the source system and target system data sharing groups? Do they have the same number of members?

1.4 Introducing the HSC test scenarios for SAP

We tested and documented two scenarios with different copy methodologies:

- ▶ Scenario 1 - Online FlashCopy with suspended DB2 in a data sharing-to-data sharing configuration
- ▶ Scenario 2 - Offline DFDSS copy/rename with restricted DB2 access in a data sharing-to-non-data sharing configuration

We used the same source system but different target systems for the scenarios.

1.4.1 Scenario 1 - FlashCopy with suspended DB2

We used the following configuration for Scenario 1:

- ▶ **Source system**

An OS/390 V2R10 Sysplex system, running DB2 V6 in a 2-way data sharing environment. We called it the RED system. SAP R/3 application server version 4.6C was installed on an RS/6000 system with a Gb Ethernet connection to OS/390 as shown in Figure 1-1.

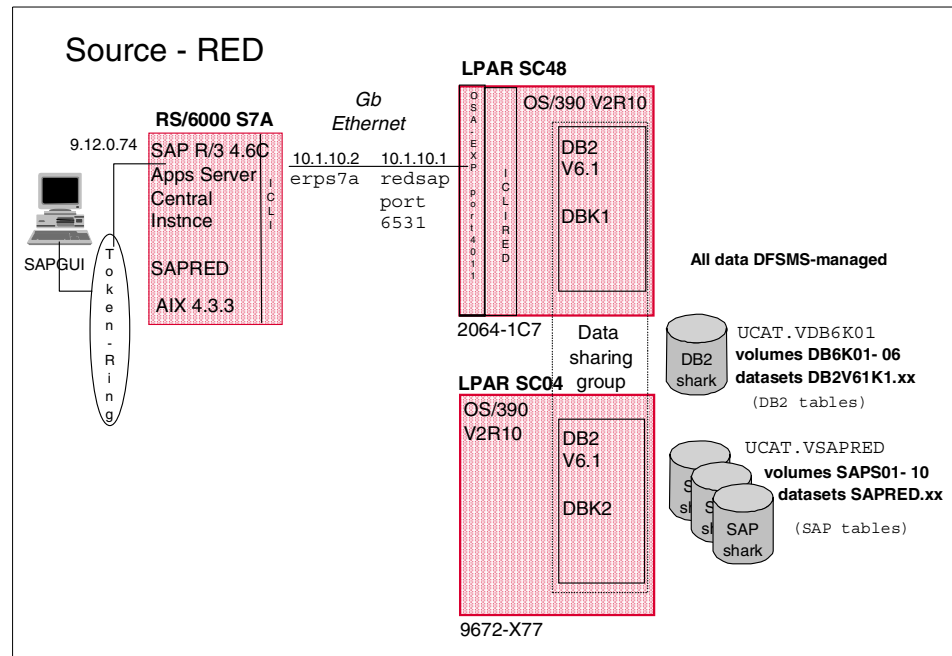


Figure 1-1 Source system in both Scenario 1 and Scenario 2

► **Rename System**

A non-Sysplex z/OS V1.2 system, without DB2. This was used for the Copy/Rename steps of the HSC process. We called it the Rename System (or Rename LPAR). See Figure 1-2.

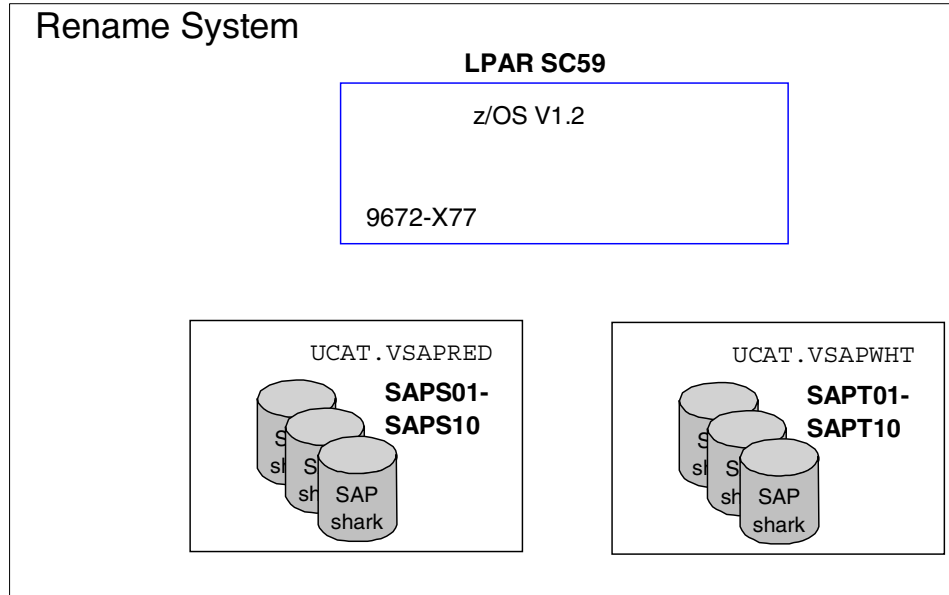


Figure 1-2 Rename System in Scenario 1

► **Target system**

An OS/390 V2R10 Sysplex system, running DB2 V6 in a 1-way data sharing environment. We called it the WHITE system. SAP R/3 application server version 4.6C was installed on OS/390 USS. See Figure 1-3 on page 12

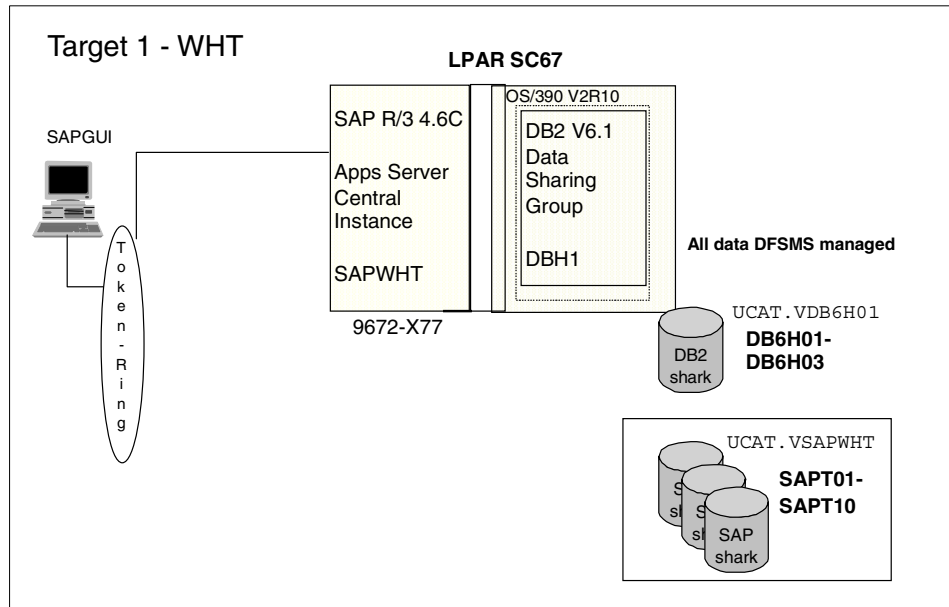


Figure 1-3 Target system in Scenario 1

1.4.2 Scenario 2 - DFDSS copy with offline DB2

We used the following configuration for Scenario 2:

► Source system

The same OS/390 V2R10 Sysplex system as for Scenario 1, running DB2 V6 in a 2-way data sharing environment (the RED system). SAP R/3 application server version 4.6C was installed on an RS/6000 with an AIX V4.3.3 system and a Gb Ethernet connection to OS/390 as shown in Figure 1-1 on page 10.

► Target system

An OS/390 sysplex system, running a non-data sharing DB2 environment. We called it the BLUE system. SAP R/3 version 4.6C application server was installed on an RS/6000 with an AIX V4.3.3 system and a Gb Ethernet connection to OS/390. See Figure 1-4 on page 13.

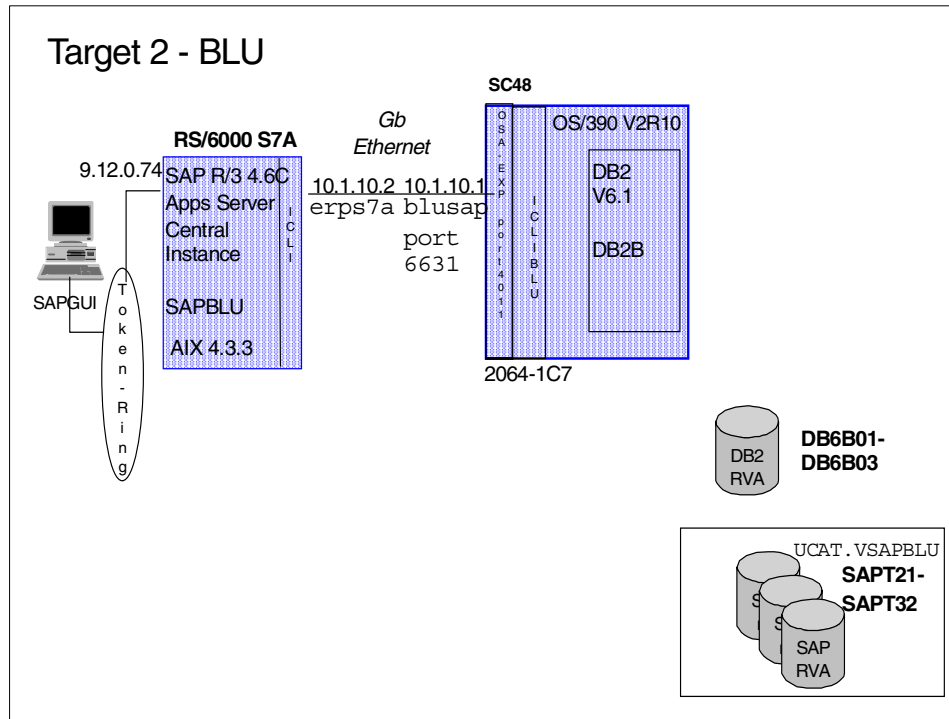


Figure 1-4 Target system in Scenario 2

1.4.3 Implementation steps for HSC process

After deciding on the methods we would use for the HSC, as described in Section 1.3.3, “Choosing a method” on page 8, which in our case were:

- ▶ Online FlashCopy with suspended DB2
- ▶ Offline DFDSS copy/rename

for each scenario, we proceeded as follows with our implementation phases:

- ▶ Planning for homogeneous system copy

In this phase of the project, we planned our resource configurations, including the maintenance we required, the disk layouts we would use, and how we would handle some BSDS issues. See Chapter 2, “Planning for DB2 system cloning” on page 15.

- ▶ Designing the homogeneous system copy process

In this phase we designed the HSC process, including configuration considerations, and the phases into which we would break the HSC process. See Chapter 3, “Designing the DB2 system cloning process” on page 23.

▶ Scenario 1 - FlashCopy with suspended DB2

Once we had planned and designed the HSC process in the given method, we executed the related HSC process. See Chapter 4, “Scenario 1 - FlashCopy with suspended DB2” on page 33. You can find the JCL in Appendix A, “JCL for Scenario 1 - FlashCopy with suspended DB2” on page 65, and the REXX execs in Appendix C, “REXX procedures” on page 105.

▶ Scenario 2 - DFDSS copy with offline DB2

Once we had planned and designed the HSC process in the given method, we executed the related HSC process. See Chapter 5, “Scenario 2 - DFDSS copy with offline DB2” on page 55. You can find the JCL in Appendix B, “JCL for Scenario 2 - DFDSS copy with offline DB2” on page 93, and the REXX execs in Appendix C, “REXX procedures” on page 105.



Planning for DB2 system cloning

This chapter describes the planning steps required before performing a DB2 system cloning.

The chapter includes planning considerations for hardware and software requirements, source copy, and target build for the methodologies discussed previously. It includes the following sections:

- ▶ Planning for online copy
- ▶ Planning for offline copy

2.1 Planning for online copy

When planning for online copy, you need to give some thoughts to the following issues:

- ▶ Timing of copying the data
- ▶ UCAT and SMS storage group configurations
- ▶ Creating a skeleton BSDS

2.1.1 Timing of copying the data

Involvement is required by the SAP administrators as to the timing of copying the data. In our scenario, the disruption was limited to the time taken to do a DB2 suspend, FlashCopy the data, and a DB2 resume. The suspend/resume function was introduced to DB2 V6.1 with APAR PQ31492 (PTF UQ36695).

2.1.2 UCAT and SMS storage group configurations

In planning for the online copying, you need to consider UCAT and SMS storage group configurations. In our test configuration, we had two UCATS for each source and target SAP instance, as follows:

- ▶ One UCAT contained the alias used by the DB2 system objects, such as DSNDB01 and DSNDB06. All of the VSAM data sets and files that make up a running DB2 subsystem were cataloged in this UCAT.
- ▶ The second UCAT had the alias used by the SAP objects. All of the thousands of VSAM data sets making up the SAP applications were cataloged in the second UCAT. The SAP VSAM data sets were managed in one SMS storage group.

With this configuration, it made the process of deleting the target SAP system very easy. The volumes in the SAP SMS pool were reclipped and the UCAT was deleted and recreated. So, it was not necessary to run thousands of VSAM deletes to remove the SAP objects. However, the DB2 system objects were removed by using VSAM deletes.

2.1.3 Skeleton BSDS for the target system

One other planning item that facilitates reduced cycle time during the target DB2 system restart is the creation of a skeleton BSDS for the target DB2 system.

During the restart, there are three elements from the source DB2 system that must be recorded in the target DB2 system:

- ▶ The source DB2 system archive log, which contains the restart point and the DB2 system checkpoint just prior to the restart point
- ▶ The DB2 system checkpoint used during restart
- ▶ The conditional restart point

When the source DB2 system archive log is registered in the BSDS of the target DB2 system, it is placed in the list of archive data sets by RBA sequence. This may not place the source DB2 system's archive log information in the right sequence. When DB2 performs the restart at the target, based on the conditional restart you provide, DB2 may find the restart point in an old and obsolete target DB2 system archive.

One method to resolve this issue is to have a BSDS for the target DB2 system that does not have any archive logs registered. This we called a *skeleton* BSDS. When the source system archive is registered in the skeleton BSDS, it will be the only one, and DB2 will find it during restart processing.

The same thing can happen with the DB2 system checkpoint. If there is an old and obsolete system checkpoint on the queue in the target DB2 system's BSDS, DB2 may reference it during restart processing. One solution is to have a target DB2 system BSDS that does not have any system checkpoints registered.

Creating the skeleton BSDS

A skeleton BSDS is a copy of the DB2 subsystem's BSDS that has all of the *registered archive log* data sets and *system checkpoints* deleted. The skeleton BSDS is only created once and kept for future cloning exercises. During the target system restart process, the skeleton BSDS is REPROed into the target DB2 system's active BSDSs to clear out old archive logs and checkpoints, prior to running the change log inventory job that registers information from the source DB2 systems.

To create a skeleton BSDS, you can use the IDCAM DEFINE JCL created at installation time. Allocate the new VSAM data set on one of the volumes with an active BSDS. See the example in Figure 2-1 on page 18.

```

//SYSIN DD *
DEFINE CLUSTER -
( NAME (DB2V61H1 . DBH1 . SKELETON . BSDS01) -
VOLUMES (DB6H03) -
REUSE -
SHAREOPTIONS (2 3) ) -
DATA -
( NAME (DB2V61H1 . DBH1 . SKELETON . BSDS01 . DATA) -
RECORDS (180 20) -
RECORDSIZE (4089 4089) -
CONTROLINTERVALSIZE (4096) -
FREESPACE (0 20) -
KEYS (4 0) ) -
INDEX -
( NAME (DB2V61H1 . DBH1 . SKELETON . BSDS01 . INDEX) -
RECORDS (5 5) -
CONTROLINTERVALSIZE (1024) ) CATALOG (DB2V61H1)

```

Figure 2-1 IDCAMS for a skeleton BSDS

Next, perform an IDCAMS REPRO of one of the active BSDSs into the skeleton BSDS. You will not be able to REPRO directly from one of the active BSDSs while the DB2 subsystem is active. Instead, REPRO the BSDS from an archive copy of the BSDS. This is the file that starts with “B”. Reference Figure 2-2.

```

REPRO IDS (SAPRES2 . DBH1 . ARCLG1 . B0000012) -
ODS (DB2V61H1 . DBH1 . SKELETON . BSDS01) REUSE

```

Figure 2-2 IDCAM control card for REPRO

Once you have a copy of the active BSDS in the skeleton BSDS, print the skeleton BSDS with the DSNJU004 program. Find all of the archive log data sets, both primary and secondary, and delete them using the change log inventory utility (DSNJU003). Also, use the change log inventory utility to cancel all DB2 system checkpoints on the checkpoint queue. *Make sure you are running the change log inventory against the skeleton BSDS.* Reference Example 2-1.

Example 2-1 Change log inventory control cards

```

DELETE DSNAMES=SAPRES2.DBH1.ARCLG1.A0000001,COPY1VOL=TOTSTU
CHECKPT STARTRBA=000000BAB934,CANCEL

```

When you have finished, run a BSDS PRINT MAP of the skeleton BSDS. After the list of active log data sets, you should see a message indicating no archive log data sets are defined. The checkpoint queue will show that no system checkpoints are recorded. Reference Example 2-2 on page 19.

Once the skeleton BSDS has been created, this action is not required for any future cloning exercises.

Example 2-2 Message in BSDS print indicating no archive log data sets are registered

```
ARCHIVE LOG COPY 1 DATA SETS  
NO ARCHIVE DATA SETS DEFINED FOR THIS COPY
```

2.1.4 Online copy steps

We used three sets of disks to exploit online DB2 system cloning using FlashCopy:

- ▶ The source system
- ▶ An intermediate system (rename system)
- ▶ The target system

This facilitated a short outage for the log suspend on the source system, and thus as little disruption to the end user as possible.

The following is a planning checklist to use with online DB2 system cloning:

- ▶ Consider differences with customizing, e.g., different release levels, OEM software, parameter settings, etc.
- ▶ Ensure that the volumes for the FlashCopy reside within the same logical control unit in the ESS.
- ▶ Ensure that there is sufficient volume at the target system for the whole DB2 environment to be copied.
- ▶ Ensure that the UCATS reside on the volumes to be copied.
- ▶ Plan a time when it is best to do a SUSPEND on the source DB2.
- ▶ Ensure that the target system UCATs and volumes are available to the rename system.
- ▶ Plan the best time for the target system to be down.
- ▶ Resolve differences between the source and target DB2s.
- ▶ Manipulate the required SAP tables to reset the ssid.

You can find a detailed description of online DB2 system cloning in Chapter 4, “Scenario 1 - FlashCopy with suspended DB2” on page 33, including the examples of JCL and REXX procedures we used.

2.2 Planning for offline copy

Your installation may not have sufficient disk available and/or may not have the necessary software/hardware to sustain an online copy, and thus might need to investigate the use of offline DB2 system cloning. This option results in increased system downtime, and has to be planned with the cooperation of the administrators of the SAP environment.

Offline DB2 system cloning typically involves copying the data to an intermediate device, such as tape, and then copying to the target system using the RENAME option. The size of a production SAP system requires that this be done during times of low utilization on the intermediate devices. Typically, a tool like DFSMS is used to move the data across the different devices.

Following is a checklist to use prior to executing the offline copy:

- ▶ Are the source and target systems at the same software levels, and will updates be required to incorporate changes?
- ▶ Is there sufficient capacity available on the intermediate devices, whether disk or tape?
- ▶ When is the best time to shut the source/target system for an extended time period?
- ▶ Can both systems be shut simultaneously and therefore bypass the intermediate copy step?
- ▶ Ensure the source/target UCAT is available where the copy will execute. This is true for both DB2 and SAP data sets.
- ▶ Ensure that the target system has sufficient disk for the entire DB2 environment.
- ▶ Are any rebinds required on the target system on completion?
- ▶ Do the parties involved in the process have the correct access to complete the required tasks?

The process will typically involve the following steps:

- ▶ Disable access to the *source* system.
- ▶ Create a Quiesce point, enforcing a recovery point.
- ▶ Shut down the *source* DB2 system.
- ▶ Back up the *source* SAP data. The number of backup jobs is determined by the availability of devices and contention with other processes being executed.
- ▶ Back up the *source* DB2 catalog/directory.

- ▶ Restart the *source* DB2 system.
- ▶ Shut down the *target* SAP app server and ICL.
- ▶ Shut down the *target* DB2 system.
- ▶ Delete the *target* SAP DB2 data sets.
- ▶ Delete the *target* DB2 catalog/directory data sets. *Do not delete* the target system BSDS data sets and active logs.
- ▶ Restore the SAP DB2 data with a rename option to the target system's HLQ.
- ▶ Restore the DB2 catalog/directory with the rename option.
- ▶ Restart the target DB2 with a COLDSTART (ENDRBA = STARTRBA).
- ▶ Consider securing the DB2 catalog/directory data sets.
- ▶ Manipulate the required SAP tables to reset the ssid.

A detailed description of the steps required to achieve an offline copy is in Chapter 5, "Scenario 2 - DFDSS copy with offline DB2" on page 55.



Designing the DB2 system cloning process

This chapter discusses the design considerations for the following system configurations:

- ▶ Data sharing to data sharing
- ▶ Data sharing to non-data sharing
- ▶ Non-data sharing to non-data sharing
- ▶ Non-data sharing to data sharing

We defined seven phases for the DB2 system cloning process, common to all methods.

- ▶ Checking the source environment
- ▶ Copying the source system
- ▶ Preparing the target disk environment
- ▶ Restoring and building target disks
- ▶ Restarting target DB2 system
- ▶ Restarting target SAP system
- ▶ Verifications and backup

Each phase has a specific deliverable. The approach taken to produce the deliverable depends on the copy method.

3.1 Configuration considerations

We consider the following system configurations when we design our copy methodology:

- ▶ Data sharing to data sharing
- ▶ Data sharing to non-data sharing
- ▶ Non-data sharing to non-data sharing
- ▶ Non-data sharing to data sharing

3.1.1 Data sharing to data sharing

In this configuration, both the source and the target DB2 systems are data sharing groups. For both *online copy* and *offline copy* methodologies, your DB2 system cloning process needs to consider the number of DB2 members in the source system and the number of DB2 members in the target system:

- ▶ In the case of online copy, the consideration is in both the process of copying the source system data sharing group and in the restart of the target data sharing group.
- ▶ In the case of the offline copy, the consideration is in the restart of the target data sharing group.

Online copy design considerations

During the online copy, all *active* DB2 members of the source data sharing group are suspended. When this copy is used to restart a DB2 data sharing group at the target, an equivalent number of DB2 members must be restarted at the target system, in order to resolve held locks. This is similar to having a DB2 data sharing member abend. In order to resolve held locks, that member must be restarted.

In order to support group restart via the coupling facility, it is necessary to have the same number of members in the target system as in the source system. However, not all of the members in the target system need to be configured as robustly as a member actually supporting a workload. In other words, the active logging configuration must be sufficient to support group restart and nothing else. The configuration to support group restart consists of each target member having BSDS data sets and the current active log from the source member available and registered in the target member's BSDS.

It may not be necessary to restart all members in the target system. If a member, or members, of the source system were quiesced and stopped at the time of the snap, these members will not need to be restarted in the target system. However, all active source members must be restarted. This is required in order to resolve local locks held by an active member. The members that are restarted will read the BSDS and the registered active logs of the members that are not restarted, and will perform group restart for these peer members.

The restart process can use active or archive logs from the source system. If the restart of the target system will be done using active logs from the source system, APAR PQ46138 is required. The active log configuration for each member of the target data sharing group can be different from that of the source system members, and different from each other.

There are many things we can change in the BSDS via the change log inventory utility (DSNJU003). However, the information about the data sharing group and its members cannot be changed. So, it is necessary to keep all BSDSs, belonging to all members, of the target data sharing group intact. That means we do not use the BSDSs from the source system to perform the restart of the target system. However, there is information in the source system BSDSs that must be recorded in the target system BSDSs in order to accomplish the restart in the target system. Depending on whether the restart is being done with the active logs versus the archive logs, the required BSDS information will vary. This information may include some, but not all, of the items below :

- ▶ The suspend LRSN, to be used as the conditional restart LRSN
- ▶ The checkpoint taken just prior to the suspend
- ▶ The archive log containing the suspend LRSN and the checkpoint
- ▶ The active log containing the suspend LRSN and the checkpoint
- ▶ The highest written RBA

To ensure the successful use of this information during the restart of the target system, you may want to consider creating a *skeleton BSDS*. See “Creating the skeleton BSDS” on page 17.

Offline copy design considerations

During the offline copy, all members of the source data sharing group are stopped. There should not be any outstanding units-of-recovery, and all data pages in the virtual bufferpools should have been externalized (written to disks). In other words, all data managed by the source system is quiesced and consistent. When this copy is used to restart the target data sharing group, a *cold start* will be performed. There is not a concern about held locks. Therefore, you can perform the restart in only one target system member.

As mentioned in the prior section, online copy methodology, we can not change the information in the BSDSs in regards to the data sharing group and its members. Again, it is necessary to keep all BSDSs, belonging to all members, of the target data sharing group intact. That means we do not use the BSDSs from the source system to perform the restart of the target system.

3.1.2 Data sharing to non-data sharing

This DB2 system cloning configuration involves moving the data from a DB2 data sharing group to a non-data sharing DB2. We tested this configuration using the offline copy methodology, so the restart of the target system would be a cold start. We tested it only using *DFDSS* copy/rename, but it is also possible to use FlashCopy to reduce the downtime of your source system.

Because the target system is non-data sharing, the DB2 system is managed by RBA and not LRSN. The target systems original BSDS and active logs are used. The information required to perform the *cold start* would be registered in the BSDSs of the target system.

In our test case Scenario 2, we used the offline copy methodology. The source DB2 system was a two member data sharing group. The target system was a non-data sharing DB2, and we took the highest used *LRSN* of our source system and used it as the restart *RBA* of our target system. Example 3-1 shows the highest used LRSN in the source system.

Example 3-1 BSDS print of source DB2 with LRSN

TIME OF CHECKPOINT	18:00:08 JUNE 18, 2001
BEGIN CHECKPOINT RBA	0012F391263C
END CHECKPOINT RBA	0012F391448C
END CHECKPOINT LRSN	B6016DA1E435

Example 3-2 shows the cold start at the target system with the source LRSN used as target start RBA.

Example 3-2 Highest used LRSN (source) to RBA (target)

```
//ACTLOG EXEC PGM=DSNJU003
//STEPLIB DD DISP=SHR,DSN=DSN610.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DB2V610B.BSDS01
//SYSUT2 DD DISP=OLD,DSN=DB2V610B.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
CRESTART CREATE,STARTRBA=B6016DA1F000,ENDRBA=B6016DA1F000
/*
//
```

3.1.3 Non-data sharing to non-data sharing

This DB2 system cloning configuration, which moves the data from a non-data sharing DB2 to a non-data sharing DB2, is described in Chapter 4 of the *SAP R/3 Homogeneous System Copy, Release 4.6C SR2* documentation. The DB2 system cloning process described in this book is for offline copy only. Should you want to use this method, this book is a good point of reference.

We did not test the online copy methodology for this DB2 system cloning configuration. If you want to use an online copy, the procedures would be a hybrid of Scenario 1 and Scenario 2.

3.1.4 Non-data sharing to data sharing

Theoretically, there is also the option of doing a non-data sharing to data sharing copy. We do not expand on this option because it is very unlikely that such a configuration will be implemented.

3.2 DB2 system cloning phases

We defined seven phases for the DB2 system cloning process, common to all methods:

- ▶ Checking the source environment
- ▶ Copying the source system
- ▶ Preparing the target disk environment
- ▶ Restoring and building target disks
- ▶ Restarting the target DB2 system
- ▶ Restarting the target SAP system
- ▶ Verifications

Each phase has a specific deliverable. The approach taken to produce the deliverable depends on the copy method.

3.2.1 Checking the source environment

This phase is intended to be completed prior to getting the copy of the source system. The deliverable of this phase is verified jobs to perform the rest of the phases. There are three different objectives in this phase :

- ▶ Gather information about the source system.

During the restart phase there are a number of alterations that are required in the target system to ensure the target system is opening and using the correct VSAM data sets. In this phase you will identify all the DB2 objects that must be altered. You will also identify all your source volumes and the copy target volumes.

- ▶ Verify jobs to perform copies and alterations

Now is the time to verify that all of the copy control statements are correct. You will also want to verify the procedures being used to ensure all the correct volumes will be online and available during the copy. If you are copying to tape, determine there are sufficient volumes and drives available. The jobs to perform the alterations should be verified. This will reduce cycle time during the restart phase.

- ▶ Create verifications

Develop a process to verify that you have copied everything required to recreate this environment in the target system. You do not want to get through the restart and find you missed pagesets.

3.2.2 Copying the source system

In this phase, the process used to get the copy of the source system is defined. The deliverable of this phase is a copy of the source system to be used in restoring and restarting the target system.

Once you have determined the copy methodology being used, it is necessary to define your copy process. The copy process should consider the following.

- ▶ Ensure the source system is in the required state prior to executing the copy:
 - Ensure there are no utilities outstanding on your source system
 - Ensure no databases or tablespaces are restricted
 - Depending on copy methodology, stop or suspend the source system
- ▶ If you are using hardware assisted volume level copy, make sure the volumes to which you are copying are online to the LPAR where the copy is being executed.
- ▶ Ensure the jobs required to get the copy will receive the resource required.
- ▶ Execute the jobs to perform the copy/rename process. At this stage, the rename would be possible only for the DFDSS process.
- ▶ Print the BSDS
- ▶ Check the steps required to resume the source system. If this is an offline copy, start the source system. If this is an online copy, resume the log.

3.2.3 Preparing the target disk environment

In this phase, the process used to prepare the target system disk for the restore process is defined. The deliverable of this phase is a target system that is ready to receive the copies of the source system through a restore process.

Consider the following:

- ▶ Establish a recovery for the target system, in case for some reason you need to get it back to where it was prior to this phase.
- ▶ Delete all of the SAP VSAM data sets. This makes these disks available to receive the copy of the SAP VSAM data sets from the source system. This includes the removal of these VSAM data sets from the OS390 user catalog.
- ▶ Delete all of the target systems directory, catalog, temporary work space, and program product VSAM data sets. There may be other DB2 system data sets that must be deleted based on the method being used to perform the restart. Again, this includes the removal of the data sets for the OS390 user catalog.
- ▶ Make the target system disks accessible from the intermediate LPAR. The OS390 user catalog for the SAP VSAM data sets and the DB2 system data sets must be known by the OS390 master catalog for this LPAR.

3.2.4 Restoring and building target disks

This phase applies to the FlashCopy process only. In this phase, the process used to copy and build the target system disks is defined. In the case of ESS FlashCopy, an intermediate LPAR becomes necessary to do the rename of the disk volumes. At the completion of this phase all of the appropriate source system data sets have been copied and renamed onto the target system disks. You need to do the following:

- ▶ Vary the FlashCoped volumes online at the intermediate LPAR.
- ▶ Import the OS390 user catalog for the source system. This includes the OS390 user catalog for the SAP data sets and the DB2 subsystem.
- ▶ Execute the jobs to perform the copy-rename process

Once the copy-rename process is complete, ensure all disks containing SAP and DB2 data sets are accessible by the target system.

3.2.5 Restarting the target DB2 system

In this phase, the process used to restart the target DB2 system is defined. The deliverable of this phase is a restarted DB2 system or data sharing group. At this point, all references to the source system will have been removed.

Once you have determined the restart methodology being used, it is necessary to define your restart process. The restart process should consider the following.

- ▶ Is the target system being restarted with the source systems active, or with archive logs?
- ▶ Is the target system being restarted with a conditional restart or a cold start?
- ▶ If this is a conditional restart, which archives will be needed to complete the restart?
- ▶ If this is data sharing, how many members are being restarted?
- ▶ Is the source system a data sharing group or non-data sharing?
- ▶ Do you have program products to consider?
- ▶ Are there maintenance level differences between the source system and the target system?
- ▶ What is the reason for this DB2 system cloning? Is it to upgrade SAP in the target system? Are there additional steps required to accomplish the upgrade?

3.2.6 Restarting the target SAP system

In this phase, the process used to restart the SAP system is defined. The deliverable of this phase is a restarted SAP application.

Execute the steps as described in Chapter 10 of the SAP Manual(SAP/R3 Homogenous System Copy, April 2001). There are some subsequent Actions for the Target System.

3.2.7 Verifications and backup

This phase defines a quality assurance process, and is being applied throughout the entire DB2 system cloning process. Each of the prior phases should have some form of verification to ensure the objectives and deliverables were met.

However, at the completion of the Restart SAP system phase, the end users should have some verification process to ensure the target system has what is required. For instance, was this DB2 system cloning performed to move application configuration information into the target system? Is the target system a production support system, and it is critical to have the right volume of data and type of data to test changes?

After the verifications have been performed, backup the DB2 and SAP systems. If you have hardware assisted volume level copy available, then use it to get a recovery copy and return the target system to the end users. However, get images copies as soon as you can. If you do not have hardware assisted volume level copy, then execute image copy jobs prior to returning the target system to the end users.



Scenario 1 - FlashCopy with suspended DB2

This chapter describes the online copy scenario we ran at the ITSO using FlashCopy with suspended DB2. We discuss copying and renaming activities, DB2 for OS/390 restart, and SAP R/3 restart. We describe the HSC process and provide guidelines to tell you what to watch for when these activities are done. We show the actual commands, JCL, and transactions used to make the copy usable.

In the online copy scenario, the restart of the target system requires a copy of the log records from the source system and information being registered in the target system's BSDSs. There are two different methods for creating the copy of the source system logs and making them known to the target system. These methods differ based on whether you are using the source system's *archive log* or the *active log* for restarting the target system.

The online copy scenario used at the ITSO used the source system's *archive log* for restarting the target system. The method for using the *active log* has also been tested, and that procedure is included.

4.1 Homogeneous system copy using FlashCopy

Online HSC using FlashCopy with suspended DB2 process is described in the following sections. This process uses the FlashCopy function of DFSS available for ESS disks. The configuration we used for this scenario is a data sharing to data sharing environment. The source system has one active DB2 member as well as the target DB2 system. We used an intermediate LPAR to copy and rename the flashcopied data sets, changing their high level qualifier into the high level qualifier of the target system. Figure 4-1 illustrates the HSC phases.

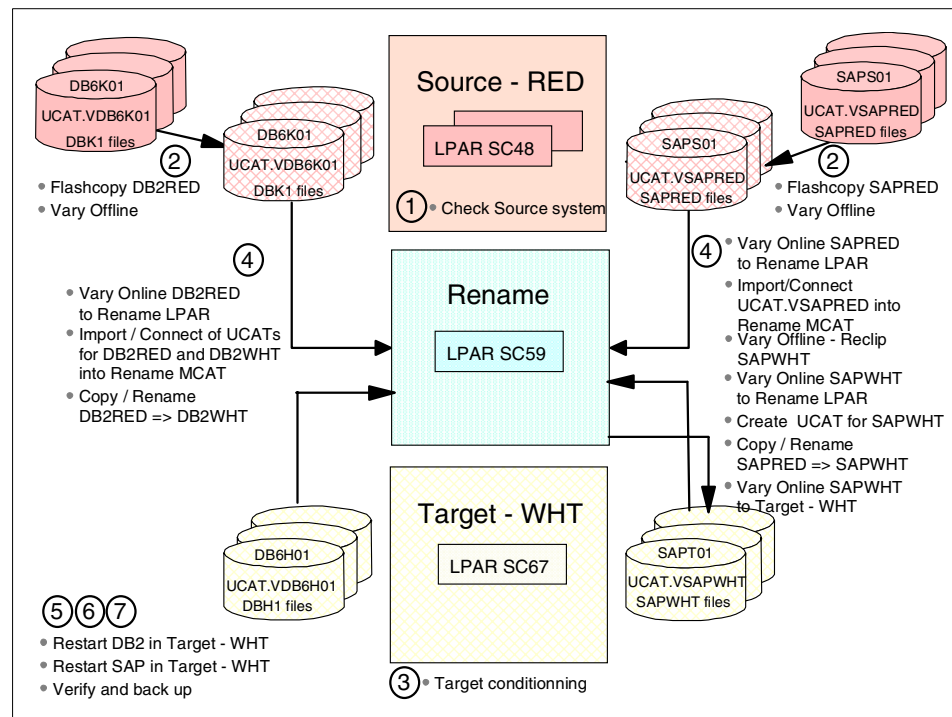


Figure 4-1 Homogeneous system copy using FlashCopy

We used the following steps in this scenario:

1. Checking the source environment
2. Copying the source system
3. Preparing the target disk environment
4. Restoring and building target disks
5. Restarting the target DB2 system
6. Restarting the target SAP system
7. Verifications and backup

4.2 Checking the source environment

This phase is used to verify the JCL that will be executed during the actual HSC process.

1. Make sure that the JCL for the FlashCopy has the correct source volumes and flash target volumes specified for:
 - a. All SAP SMS pool volumes
 - b. DB2 directory and catalog volumes
 - c. Any volume with vendor packages on them
2. Check the volumes to receive the FlashCopy data are online to the system on which the copy will be executed. On completion of the copy, the target packs will be varied offline by FlashCopy. This is to prevent having duplicate volume names on the same LPAR.

FlashCopy has been updated to include the *dumpconditioning* parameter that does not vary the target volumes offline on completion of the copy. The target volumes are kept online with the original volser but the data sets are not cataloged. For information please check APAR OW45674.

We saw no advantages if using the *dumpconditioning* parameter in the cloning process. The volumes would still be required to be varied offline and clipped to clear the SAP datasets. A reason to use this option, would be to have no duplicate volsers on completion of the *FlashCopy*.

3. Using ISPF option 3.4, list and record the number of DB2 system data sets to be copied. Typically, there are too many volumes for the SAP VSAM data sets. This check is *only* for DB2 system VSAM data sets, program product VSAM data sets, and any system libraries that need to be copied. The following table is an example of recording the information to use for checking copied data sets.

Table 4-1 Source DB2 production data set counts

SOURCE PRODUCTION VOLSER	DATASET COUNT	FLASH-TARGET
DB6K01	16	DB6K07
DB6K02	25	DB6K08
DB6K03	43	DB6K09
DB6K04	33	DB6K10
DB6K05	24	DB6K11

SOURCE PRODUCTION VOLSER	DATASET COUNT	FLASH-TARGET
DB6K06	22	DB6K12

4. Review the DFDSS COPY/RENAME jobs to make sure the JCL in the SYSIN DD is correct.
5. Verify the release and maintenance levels of the DB2 subsystem, and all program products, as currently installed in the source and the target systems. Determine if the maintenance levels will change via the homogeneous copy process. Identify additional steps required to enable new DB2 features or program products in the target system.

4.3 Copying the source system

In many installations, copying the source system can be quite a challenge. In order to provide high availability, the source system must stay up and be available to its end-users. With the DB2 SET LOG SUSPEND command, we can stop the update process long enough for DB2 to record the required information needed by restart to establish data consistency. However, there could be certain activities running in the DB2 data sharing group that would force us into object recovery when we restart at the target system. It is necessary to check the DB2 source system to ensure none of these activities or status exist. Once it has been established, we want to suspend the active log activity, obtain our FlashCopy and resume the active log as quickly as possible.

It is recommended that these steps be automated, if possible. This is to ensure minimal suspend time during the FlashCopy process.

Sample JCL jobs from Appendix A, “JCL for Scenario 1 - FlashCopy with suspended DB2” on page 65 are indicated by “{name}”. Copying the source system involves the following steps:

1. In order to reduce the amount of time that the DB2 active logs are suspended, the FlashCopy job must execute very quickly. The following steps must be executed prior to suspending the DB2 active logs:
 - a. On the target system, EXPORT DISCONNECT the target SAP UCAT.{Example A-4}
 - b. On the intermediate LPAR, EXPORT DISCONNECT the target and source systems SAP UCAT. At the same time, EXPORT DISCONNECT the source DB2 UCAT. {Example A-4}
 - c. Vary the FlashCopy receiving packs offline across all LPARs.

- d. On the intermediate LPAR, clip the volumes from the preceding step, to re-establish the original names and to remove any datasets that may reside on these volumes. Multiple jobs can be created and executed in parallel. The recommended size for the VTOC is 400 tracks, VTOC index is 50 tracks, and the VVDS is 180 tracks.{Example A-5}
 - e. Vary the clipped volumes online to the source DB2 LPAR.
2. Disable the ability for utility jobs to start.
This could be accomplished by assigning utility jobs a specific initiator and draining them.
 3. Ensure no utilities are running or registered in SYSUTILX. Make sure no pagesets are in a restricted status.

```
DISPLAY UTIL(*)  
DISPLAY DATABASES(*) SPACE(*) LIMIT(*) RESTRICT
```


This step can run for a few minutes because of the large number of SAP objects. However, at this point we have not suspended the active log, so length of time is not a concern.

If there are any utilities registered in SYSUTILX or any pagesets are in a restricted status, do not execute the FlashCopy. Resolve the outstanding utility and the restricted pagesets, then continue.
 4. The following steps are meant to be executed quickly in order. Once the DB2 active log has been suspended, complete the FlashCopy as quickly as possible and resume activity to the DB2 active logs. The recommendation is to resume the DB2 active logs at the completion of the FlashCopy, whether or not the FlashCopy ended successfully. If the FlashCopy did not complete successfully it is necessary to recheck that no utilities are running or there are pagesets in restricted state.
 - a. Force DB2 system checkpoint across all members of the data sharing group.

```
SET LOG LOGLOAD(0)
```
 - b. Execute log suspend on all members of the data sharing group.

```
SET LOG SUSPEND
```


This is not a data sharing group wide command and requires to be executed against each active member of the source data sharing group.
 - c. Print BSDS for source data sharing group.{Example A-2}

The purpose for this BSDS print is to easily provide information for steps documented in the DB2 system restart procedure.

Depending on whether you are using the source system's archive or active logs for restarting the target system, different fields of information in this BSDS print will be used.

If the source system's archive logs are being used to restart the target system, the information for the suspend checkpoint, as reported in the DSNJ372I message, will be required. The suspend LRSN will be obtained from the DSNJ372I message issued by DB2 when the SET LOG SUSPEND command was issued. If the source system's active logs are being used, then the "highest RBA written", its timestamp, the "highest RBA offloaded", and the current active log information will be used. It is important that the BSDS print be obtained prior to resuming the log. So, save this print map and record where it is saved.

Table 4-2 BSDS print map with log suspend information

BSDS PRINT JOB NAME	BSDS PRINT JOB NUMBER
SAPRES3A	JOB08736

- Execute the FlashCopy JCL. {Example A-3 on page 66}

The PARALLEL keyword we used in the FlashCopy syntax speeded up the execution of the logical copy.

- Resume the log on all members of the data sharing group.

```
SET LOG RESUME
```

- Find the DSNJ372I message created by each source DB2 member. In this message is the STARTRBA for the system checkpoint that was forced prior to the suspend. Find this checkpoint on the checkpoint queue in the BSDS print (Table 4-2). Run a DSN1LOGP job supplying the STARTRBA and the ENDRBA of the checkpoint. Look for long running units-of-recovery (UR).

It is necessary to understand how many and which source active or archive log data sets will be required to recover the long running UR during the restart of the target DB2 system. Record this information to be used during restart preparation.

4.4 Preparing the target disk environment

This section describes the steps required to be completed prior to replacing the target volumes by the Copy/Rename step. Back up the environment before completing these steps, and once done, the target system will be destroyed. The JCL jobs mentioned are listed in Appendix A, "JCL for Scenario 1 - FlashCopy with suspended DB2" on page 65.

1. Before starting this procedure, make sure any SAP application monitor alarms are disabled on the target system.
2. All production jobs, utility or SAP application, should be placed in 'non-execute' mode.
3. Shutdown ICL and SAP application servers
4. Force archive across all members
-ARCHIVE LOG MODE(QUIESCE)
5. Shut down all DB2 members for the target DB2 data sharing group.
6. If hardware assisted volume level copy is available, create a copy of the target DB2 system. This would include the DB2 system objects, DB2 system libraries, all SAP DB2 objects, and any program product libraries and DB2 objects.
7. Validate the number of data sets being restored to each target volume. This is only done for the non-SAP data volumes. Compare the counts recorded for the source volumes in Table 4-1 on page 35 to the counts on the target volumes. To find the numbers, use ISPF option 3.4 with DSNAME LEVEL-"high-level-qualifier"; VOLUME SERIAL- "production target pack".

Table 4-3 Compare source with target system data set count

SOURCE VOLSER	SOURCE COUNT	TARGET VOLSER	TARGET COUNT
DB6K01	16	DB6K07	16
DB6K02	25	DB6K08	25
DB6K03	43	DB6K09	43
DB6K04	33	DB6K10	33
DB6K05	24	DB6K11	24
DB6K06	22	DB6K12	22

8. Get BSDS print of target DB2 data sharing group. Save and store this print. If it is necessary to perform a recovery of the target system, this BSDS print may be used.{Example A-9}
9. You will need to use DB2 native stop and start commands. If you have your DB2 started tasks under automation control to stop and start DB2, remove them from automation control now.
10. Start DB2 in maintenance mode:
=DBH1 START DB2 ACCESS(MAINT)

11. Logon as one of the install SYSADMs.
12. Drop the DSNR* objects as per the install job DSNTIJSJ. Expect a return code of "8". {Example A-11}
13. Drop all of the DSNR* objects on the target system. The image copy data sets created by the DB2 IVP jobs need to be included in this job. {Example A-12}
14. If you have program products that are using DB2 STOGROUP defined data sets, run a job now to drop the databases. If there are any other data sets used by the program product, such as control or history files, this would be a good place to delete them, as well. These data sets will be brought over by the HSC process.
15. Shut down the target DB2 system and any associated members.
=DBH1 STOP DB2
16. Vary the target volumes offline across all LPARs, ensuring the volumes are offline on the intermediate LPAR. Re-clip the volumes in the SAP SMS pool, and delete and recreate the UCAT for the SAP target VSAM datasets on the intermediate LPAR. {Example A-8}. On completion vary the volumes back online across all LPARs.
17. Set up and run the job to delete VSAM clusters for the indexes created on DSNDB06 by the SAP installation process. {Example A-10}
18. Delete all the VSAM data sets for the DB2 directory, catalog, temp work tablespaces, RUNLIB and SRCLIB data sets. {Example A-13}

Attention:

It is critical that the target BSDS, active log data sets, and DBRMLIB do not get deleted because these are not FlashCopy'ed from the source system.

19. Depending on if you archive to disk or tape, you need to either delete or uncatalog the archive data sets. Set up and run jobs to delete or uncatalog all archive data sets for all data sharing members. If you archive to disk and do not have a tape copy of the archives, it would be a good idea to backup the last two archives to tape prior to deleting them from volume. This is in case you need to recover the target system. {Example A-14}

Using ISPF option 3.4 enter the high level qualifier to list all archive data sets
DB2V61H1.DBH1.ARCLG*
20. If each DB2 subsystem has its own copy of DB2 system libraries, and various members in these libraries have been customized to a specific DB2 subsystem, rename the current target DB2 subsystem's libraries for *all* data sharing members. For example, if you customize panels in DSNPFPP, you will

need to copy those members back into the newly allocated DSNPSP libraries as part of completing the target DB2 system restart process.

If you only have one set of DB2 system libraries for each maintenance level of DB2, then it will only be necessary to determine which level will be appropriate for the target DB2 system.

21. From ISPF 3.4 list all data sets that use the vcat of your target DB2 system - DB2V61H1. The only remaining entries should be the following:
 - Catalog alias
 - BSDSs for all members
 - Active log data sets for all members
 - Skeleton BSDSs
 - DBRMLIB
 - Possible: SMP/E libraries
 - Possible: DB2 system libraries

4.5 Restoring and building target disks

On successful completion of the FlashCopy, the following steps are required to enable access to the data that has been copied from the source system. The time lapsed is dependant on ones requirements. Bear in mind that the volumes that have received the FlashCopyFlashCopy data are offline, but with the same volser as the originals.

1. VARY ONLINE to the intermediate LPAR, the DB2 and SAP volumes that received the FlashCoped data. As shown in Example 4-1, it is possible to create a member in SYS1.PROCLIB to execute these commands. At a site with large number of volumes, this will speed up the process.

Example 4-1 SYS1.PROCLIB member to VARY volumes ONLINE

```
//SAPWHTON PROC
//*
// COMMAND 'RO *ALL,V 6200,ONLINE'
// COMMAND 'RO *ALL,V 6201,ONLINE'
//* STARTUP PROCEDURE FOR VARYING SAP VOLUMES ONLINE
//*
//ONLINE EXEC PGM=IEFBRI4
```

2. Make sure that target volumes are ONLINE to intermediate LPAR. These are the target DB2 catalog/directory, SAP, and any active log volumes being copied.

It is up to each installation using the source active logs for restart, as to which source active logs to copy over to the target systems log volumes. Unless there was a long running unit-of-recovery (UR), it is only necessary to copy the active logs required to restart the target system. Typically, this would be the source systems current active log. However, the system checkpoint reported in the DSNJ372I message must be included on this log.

3. IMPORT CONNECT and DEFINE ALIAS of the source system to the intermediate system's Master Catalog {Example A-6}.
4. Define a UCAT for the target SAP data sets. This would have been deleted by step 4.4, "Preparing the target disk environment" on page 38. See Example A-7.
5. Copy the data from the intermediate volumes to the target volumes. Use the RENAME parameter to change the high-level-qualifier from the source system to the target system. Create several jobs that can execute in parallel and thus reduce the time required {Example A-28}.
6. If necessary, reallocate non-VSAM DB2 data sets for the target subsystem. Copy the source system libraries into these newly allocated libraries. If DSNLOAD is one of these, verify that the target library is allocated on an APF authorized volume.
7. If necessary, copy members from the target's old system libraries, that were renamed in Step 20. on page 40, to new target libraries.

4.6 Restarting the target DB2 system

Remember that the BSDSs are not FlashCopy'd from the source DB2. If you are using source system active logs to restart the target DB2 system, you will FlashCopy active logs over to the target DB2 system.

The first section of the restart process involves collecting restart information that will be used when restarting the target DB2 system. This information will be obtained from the source DB2 system. It will vary based on whether the source system's active or archive logs are being used to restart the target system.

The second section describes the steps required to restart the target DB2 system. This includes:

- ▶ Ensuring that the appropriate source archive or active logs are available on disk at the target DB2 system.
- ▶ Making DB2 system and program product STOGROUP VCAT changes.
- ▶ Using the change log inventory job to empty out the active logs, register the source archive or active logs at the target, provide a system checkpoint RBA,

and establish the restart point. The restart point will be established either by a conditional restart, if using archive logs to restart the target system, or by changing the “highest RBA written” when using active logs to restart the target system.

The third section discusses possible customizations that may need to be completed, depending on your installation.

4.6.1 Collecting restart information for the target DB2 system

You can restart the DB2 system using either active logs or archive logs. In each case you need first to collect some restart information.

Active log restart information

If you are using the active logs to restart the DB2 system, you need to execute the following steps to collect restart information.

1. For each source member, locate the DSNJ372I message that was generated when the suspend was executed. If the source DB2 systems have not been restarted, this message can be found in the JESMSGLOG for MSTR. If they have been restarted, go to the SYSLOG, for the appropriate LPARs, on the date of the suspend execution. Record the information in Table 4-4.

Table 4-4 Active log - Information given in the DSNJ372I message

MEMBERS	SUSPEND RBA	SUSPEND LRSN	SUSPEND CHKPT RBA
MEMBER1	001293DC0B77	B5FC2E8726FF	001293658EB6
MEMBER2

2. Locate the BSDS print output from the job recorded in Table 4-2 on page 38.
3. For each member, identify the active logs required to restart the target system. Again, if there was a long running UR more than just the current active logs will be required. Record the active log information in Table 4-6 on page 45.
4. For each member, find in the BSDS print the “highest RBA written”, the timestamp, and the “highest RBA offloaded”. Record this information in Table 4-7 on page 45.

Archive log restart information

If you are using the archive logs to restart the DB2 system the following steps must be completed for all active source DB2 system members.

1. For each source member, locate the DSNJ372I message that was generated when the suspend was executed. If the source DB2 systems have not been

restarted, this message can be found in the JESMSG LG for MSTR. If they have been restarted, go to the SYSLOG, for the appropriate LPARs, on the date of the suspend execution. Record the information in Table 4-5.

Table 4-5 Archive log - Information given in the DSNJ372I message

MEMBERS	SUSPEND RBA	SUSPEND LRSN	SUSPEND CHKPT RBA
MEMBER1	001293DC0B77	B5FC2E8726FF	001293658EB6
MEMBER2

2. Locate the BSDS print output from the job recorded in Table 4-2 on page 38.
3. For each member, do a FIND in the BSDS print for the suspend chkpt RBA recorded in Table 4-5 on page 44.

```

TIME OF CHECKPOINT13:48:38 JUNE 14, 2001
BEGIN CHECKPOINT RBA001293658EB6 (SUSP CHKPT RBA)
END CHECKPOINT RBA001293682A22
END CHECKPOINT LRSNB5FC2DF54747

```

4. Record the checkpoint information in Table 4-10 on page 47.
5. Record suspend LRSN from Table 4-5 in Table 4-11 on page 47.
6. The archive log for system restart is not included on this BSDS print. Run a DSNJU004 job on the source system.
7. Look for archive log that contains the suspend LRSN. If the suspend LRSN has not been archived, force an archive across all members of the group. Rerun the BSDS print job.
8. Record the archive log DSN and VOLSER in Table 4-8 on page 46
9. Record the archive log STARTRBA, ENDRBA, STRTLRSN, and ENDLRSN information in Table 4-9 on page 47. The keywords FORWARD and BACKOUT should both equal YES.

4.6.2 Restarting the target DB2 system

After collecting restart information you are ready to restart the target DB2 system. You can do this using either DB2 active logs or archive logs.

Restart using DB2 active logs

All jobs to manipulate the BSDS for the target DB2 system *must* be completed for all target DB2 members. The target system restart varies depending on how many members are in the target DB2 data sharing group.

1. Repro the skeleton BSDS to the target system BSDS's {Example A-16.
2. Edit and run the change log inventory jobs for all members of the target system. Each change inventory job should have three steps; redefining the

active logs, re-establishing the “highest RBA written”, its timestamp, and the “highest RBA offloaded”, and creating a conditional restart control statement. The change log inventory job will end with a condition code of 04, because the current active logs for the target system are being deleted. Ensure each individual step completes successfully {Example A-17 on page 77}.

- a. All of the active logs will be deleted and redefined using the DELETE and NEWLOG control statements. The only active logs needing the STARTRBA and ENDRBA are the ones required to successfully restart the target system. Again, typically these would be only the current active logs from the source system. However, if there was a long running UR, more may be required. Once this change inventory step has completed, most of the active logs in the target system should be reported in the BSDS as empty, reusable.

Table 4-6 Active log information taken from BSDS print

MEMBERS	ACTIVE LOG DSN	STARTRBA	ENDRBA
MEMBER1		001293DB3000	0012A1658DFFF
MEMBER2

- b. The next step in the change inventory job is to re-establish the restart RBA. This is accomplished by using the HIGHRBA control statement and providing the “highest RBA written” using the the STARTRBA parameter. The time parameter is also required. Use the OFFLRBA parameter to re-establish the last RBA archived. This is required to ensure the DB2 system will archive the active logs when full.

Table 4-7 Restart RBA information from BSDS print

MEMBERS	HIGHEST RBA WRITTEN	TIMESTAMP	HIGHEST RBA OFFLOADED
MEMBER1	001293DC0B77	2001.159 06:04:04.0	001293DB2FFF
MEMBER2

- c. The third step uses the conditional restart control statement. The only reason for this CRESTART statement is to provide DB2 the startdba of the checkpoint completed prior to the suspend. This is done with the CHKPTRBA parameter. Use the SUSPEND CHKPTRBA recorded in Table 4-4 on page 43. We do not want to truncate the log, so none of the parameters to provide truncation RBAs will be used.
3. Once the change log inventory job has completed successfully for all target DB2 members, run a BSDS print map for all members. {Example A-19}

- Validate each active log. Make sure the startbba and endrba for the ones required for restart are correct. The rest of them should be EMPTY, REUSABLE.
- Validate the “highest RBA written”, its timestamp, and the “highest RBA offloaded” .
- Validate the conditional restart card. Make sure there is no truncation point. Verify that the CHKPTRBA is correct.

Restart using DB2 archive logs

All jobs to manipulate the BSDS for the target DB2 system *must* be completed for all target DB2 members. The target system restart will vary depending on how many members are in the target DB2 data sharing group. Table 4-8 lists the source DB2 systems archive logs to be registered in the target system’s BSDS.

Table 4-8 Source system archive data sets to be used in restart of target system

MEMBERS	SOURCE SYSTEM ARCHIVE DSN	VOLSER
MEMBER1	SAPRES2.DBK1.ARCLG1.A0000061	TOTTSQ
MEMBER2

1. Edit job to restore the archive log from the source DB2 system. This job uses IEBGENER to make a volume copy of the source DB2 system archive that is used by the target DB2 member for restart. {Example A-15}
 - a. SYSUT1 will be the source system archive DSN Table 4-9 on page 47
 - b. SYSUT2 will be the target system archive DSN. The sequence number will always be A0000001. Record the volser of this data set as COPY1VOL in Table 4-9 on page 47

If you identified long running URs on the source DB2 system (Step 7. on page 38), the appropriate source system archive log data sets must also be restored to disk.

2. Repro the skeleton BSDS to the target system BSDSs {Example A-16}.
3. Edit and run the change log inventory jobs for all members of the target system. The change log inventory job will end with a condition code of 04, because the current active logs for the target system are being deleted. Ensure each individual step completes successfully {Example A-18}.
 - a. All of the active logs will be deleted and redefined without the STARTRBA and ENDRBA. They should show as empty, reusable in the BSDS.

- b. The target BSDS needs to be updated with the archive data set information that was restored to disk in Step 1. on page 46. Table 4-9 lists the information required for change log inventory utility DSNJU003.

Table 4-9 Source DB2 system archive log data information

	MEMBER 1	MEMBER 3
STARTRBA	00128A6DE000
ENDRBA	0012B65FDFFF
STRTLRSN	B5FB4F6963A0
ENDLRSN	B5FC3A42AF AF
COPY1VOL	TOTTSQ
UNIT	3390
CATALOG	YES	YES

- c. Add the system *restart checkpoint* to the target system's BSDSs. This is done using the change log inventory CHECKPT statement. Table 4-10 contains the information required for the checkpoint. This is required as we use a skeleton BSDS and thus need to register the required restart information as extracted from the source system.

Table 4-10 Source DB2 system checkpoint information

	MEMBER 1	MEMBER 2
STARTRBA (Use in CRESTART as CHKPTRBA)	001293658EB6
ENDRBA	001293682A22
ENDLRSN	B5FC2DF54747
TIME (YYYYDDDDHHMMSS)	2001165134838

- d. Create the *conditional restart*. This is done using the change log inventory CRESTART statement. The ENDLRSN is the suspend LRSN. Make sure the CHKPTRBA equals the STARTRBA of the checkpoint in Table 4-10. Table 4-11 lists the information required for the conditional restart.

Table 4-11 Conditional restart information

MEMBER	SUSPEND LRSN	CHKPTRBA
MEMBER1	B5FC2E8726FF	001293658EB6
MEMBER2

4. Once the change log inventory job has completed successfully for all target DB2 members, run a BSDS print map for all members {Example A-19}.
 - Validate that each active log is EMPTY, REUSABLE.
 - Validate the added archive for each member.
 - Validate the conditional restart card.
 - Validate the checkpoint that was added.

Remaining restart steps using either active or archive logs

The remaining restart steps are as follows:

1. Create special startup ZPARMs for the target DB2 member being used for completion of the homogeneous copy {Example A-20}:
 - Change RESTART to DEFER.
 - Change SYSADM2 to the ID of the DBA that will be performing the *alter-drop/create-alter* work for the SAP objects.
 - Make sure the right SMP/E zone libraries are being used.
2. For the remaining members of the target DB2 system, do the following:
 - Change RESTART to DEFER.
 - Make sure the right SMP/E zone libraries are being used.
3. Prior to starting, the SCA and LOCK structures must be deallocated in the coupling facility.

Important:

Do this prior to every restart attempt:

- a. Issue the following MVS commands.

```
SETXCF FORCE, CONNECTION, STRNAME=DB2V61H1_LOCK1, CONNAME=ALL
SETXCF FORCE, STR, STRNAME=DB2V61H1_LOCK1
SETXCF FORCE, STR, STRNAME=DB2V61H1_SCA
```

- b. Issue the following MVS command to validate that all of the coupling facility structures for this data sharing group have been “not allocated”. This will force it into group restart.

```
D XCF,STR
```

Do a **find** for the group name to get to the block of output showing the status of the applicable structures. All should be “not allocated”.

4. Start the target DB2 subsystem:
 - a. Start the first target DB2 member.

```
=DBH1 START DB2 ACCESS(MAINT)
```

- b. Before replying to the DSNJ245I message, verify the conditional restart value with the ENDLRSN value in Table 4-11 on page 47.
- c. If there is more than one member in the target system, the following message appears:

```
nn DSNR020I =ssid DSNRRUTT START MEMBER ssid, OR REPLY 'NO' OR 'QUIESCED'
```

Start the remaining target DB2 member(s).

```
=ssid START DB2 ACCESS(MAINT)
```

- d. Analyze the startup. Do not proceed if the target DB2 member(s) start with errors or have units of work needing recovery. An incorrect value for the CRESTART may cause this type of problem.
- e. After all members have successfully started, stop *all* members not being used in completion of homogeneous copy.

```
=ssid STOP DB2
```

5. Issue a start database command for any DB2 directory or catalog pagesets. Normally, SYSLGRNX and its indexes will have LPLs.

```
=DBH1 START DB(DSNDB01) SPACE(*)
```

```
=DBH1 START DB(DSNDB06) SPACE(*)
```

Verify the pagesets in DSNDB01 and DSNDB06 to ensure none are restricted.

```
=DBH1 DISPLAY DB(DSNDB01) SPACE(*) LIMIT(*) RESTRICT
```

```
=DBH1 DISPLAY DB(DSNDB06) SPACE(*) LIMIT(*) RESTRICT
```

6. Drop the temp workspace database. Note that the data base names are for the source system. Do not change these names. The SAP indexes that are created against the DSNDB06 are ALTERed at the same time {Example A-21}.
7. Create the temp workspace database for the target system {Example A-22}.
8. Extract the DB2 objects that need to be ALTERed, STOPped and STARTed. These would be the DB2 system and SAP database, tablespaces and indexes. Examples of REXX procedures (Appendix C, "REXX procedures" on page 105) and JCL to execute these are included in Example A-23 on page 83.
9. If your SAP database is 4.6C or higher, it might be necessary to increase the size of the SCA. We found an allocation size less than 24576K was too small during the process that changes the vcat for the SAP objects.

- First, display the CF structure

```
D XCF,STR,STRNAME=DB2V61H1_SCA
```

This shows:

```
POLICY SIZE      : 32768K          <--- MAX
POLICY INITSIZE: 16384K
ACTUAL SIZE      : 16384K          <--- Current
```

- If necessary, issue this command to increase the actual size.

```
SETXCF START,ALTER,STRNAME=DB2V61H1_SCA,SIZE=32768
```

10. Start the *alter-drop/create-alter* process for the SAP objects. This process will run for a number of hours {Example A-24}.
11. Make sure all of the DB2 system STOGROUP managed object's VCAT have been changed to the target system VCAT. Validate that the SAP VCAT names have been ALTERed correctly. In our testing this was DB2V61K1 to DB2V61H1 {Example A-25}.
12. Remove the target SAP UCAT and the SOURCE SAP UCAT and DB2 system UCAT from the rename LPAR {Example A-4}.
13. Vary the target volumes offline from the intermediate LPAR.
14. Vary the volumes online to the target system, this been the system where the cloned DB2 executes.
15. Import the SAP UCAT {Example A-26}:
 - The SAP UCAT will reside in the target LPARs master catalog and needs to be removed with the EXPORT DISCONNECT command
 - Re-establish the SAP UCAT with the IMPORT CONNECT command and recreate the alias.
16. Recompile the ZPARM module. Use the setting for the parameters that would normally be used. Make sure the right level SMP/E zones are being used. The target system is now at the same level as the source system. Repeat for all target data sharing group members {Example A-27}.

If necessary update any reference to the SMP/E zone for each target system data sharing group member.
17. Take target system down

```
=DBH1 STOP DB2
```
18. If you have your DB2 started tasks under automation control to stop and start DB2, move them back into automation control now.
19. Reset the size of the SCA back to the init size -

```
SETXCF START,ALTER,STRNAME=DB2V61H1_SCA,SIZE=16384
```
20. Start up the target system. Check for any *restricted* DB2 objects.

```
DISPLAY DATA BASES(*) SPACE(*) LIMIT(*) RESTRICT
```
21. If part of a data sharing DB2 group, reset the outstanding members.

22. Vary the intermediate volumes offline and re-clip to the original names. This resolves the problem of having duplicate volumes when any system is IPLed. See section 4.3, “Copying the source system” on page 36 {Example A-5}.
23. Turn over to BASIS to complete manipulating the SAP references from the source SID to the target SID.

4.6.3 Installation customizations

The following work is very installation dependent. The following steps are intended to suggest possible changes that may be required. Each installation needs to give thought to their individual naming standards and configurations relative to DB2 and program products. In order to avoid contention, it is recommended the following steps *not* run concurrently with the process for changing the vcat for the SAP objects.

- ▶ Free plans that reference back to source system. If you use the individual *ssids* of the DB2 subsystems in the plan names for SAP or any other product, free the plan names belonging to the source system.
- ▶ Validate name of the default plan to be used by ICLI server(s) for target DB2 system(s), by looking at the PROC for each started task. If your naming conventions for the SAP plans uses the *ssid* of the individual data sharing group members, this needs to be considered when binding the plans for the target system.

Make sure all appropriate packages are listed in PKLIST.

- ▶ Perform GRANTS for ICLI server(s) to target DB2 system
- ▶ If the SAP release is less than 4.5 (pre-RRSAF), you may need to bind multiple plans.

Remember to grant all plan names to the ICLI server(s).

- ▶ Revoke GRANTS for source system ICLI server(s) and source system program product user names.
- ▶ Find and alter all stored procedures that are using the WLM environment.
- ▶ Determine if version or maintenance level for any program products has changed in the target system. If the source system and target system were at different levels, it is necessary to re-create all the program product system libraries for the target system.
 - Rebind the program product packages and plans
 - If program product is being up-leveled on the target system, via the homogeneous copy, verify the need for maintenance to customized parameters

- ▶ In all cases, rerun the program product GRANTs for all program product users belonging to the data sharing group.

4.7 Restarting target SAP system

Before starting the SAP system, you must install the SAP license. If the target SAP system message server is running on the same hardware as the original install, then you can use the same SAP license that you obtained originally from SAP. If not, you have to obtain a new SAP license from SAP.

You install the the SAP license by logging on to the message server and issuing the following command and supplying the necessary information at the prompts:

```
saplicense -install
```

You should also delete the license for the source system which was copied during the homogeneous system copy process. You do this by entering the following command and supplying the necessary information at the prompts:

```
saplicense -delete
```

Also, before starting the SAP system, you must ensure that the DB2 dual logging and archiving are enabled. In our scenarios we turned archiving off while we were altering the DB2 catalog. After we completed altering the DB2 catalog we stopped DB2, reassembled the DSNZPARM, and restarted DB2 in normal mode.

Prior to starting the SAP system, you must review the chapter on General Subsequent Actions of the *SAP R/3 Homogeneous System Copy, Release 4.6C SR2* manual. This chapter documents actions to be taken using native DB2 processes like deleting all rows from a list of SAP tables. It further documents actions to be taken using SAP R/3 transactions like configuring the Workbench Organizer.

If the target system has a SAP application server which is at a different release or executes on a different platform, do the following steps.

- ▶ Bind packages and plan to be used by SAP appserver.
- ▶ Perform grants for use of this plan and additional privileges for sidADM.

4.8 Verifications and backup

It is necessary to verify the whole system. There were different points throughout the process where validations were run to make sure there is no reference back to source system VSAM data sets or configuration. However, it is recommended that each installation develop its own SAP application and program product verifications.

These verifications should take into consideration the reason for the homogeneous copy. For instance, this may have been done to bring a SAP application configuration change into the target system. This should be verified. Maybe the source system had business data in it that needed to be used in a certain test process. Validate that the data is as you need it.

Once the restart of the target SAP system is complete, the SAP/DB2 environment must be backed up. The image copy job for the DB2 system objects runs quickly. However, to image copy the SAP objects can take several hours. In order to turn the target system over to the end-users quickly, run the image copy job for the DB2 system objects, and if hardware assisted volume level copy is available, create a copy of the target DB2 system. This will become the first recovery point for the newly created target system.

Get current image copies of the SAP objects after returning the target system to the end users. This can be done via the normal job schedule for image copy jobs.



Scenario 2 - DFDSS copy with offline DB2

This chapter describes the process of homogeneous system copy using DFDSS copy/rename with an offline DB2. Many steps of this process are similar to the online copy, with the major difference being that the target DB2 is *cold started* and thus no recovery is done. This scenario discusses the cloning of a data sharing SAP/DB2 system to a non-data sharing environment. In this scenario, the source system is unavailable when the copy/rename jobs are executing.

5.1 Homogeneous system copy using DFDSS copy

The following sections describe an offline HSC using DFDSS copy/rename with restricted DB2 access.

Several options are possible in this scenario, as shown in Figure 5-1:

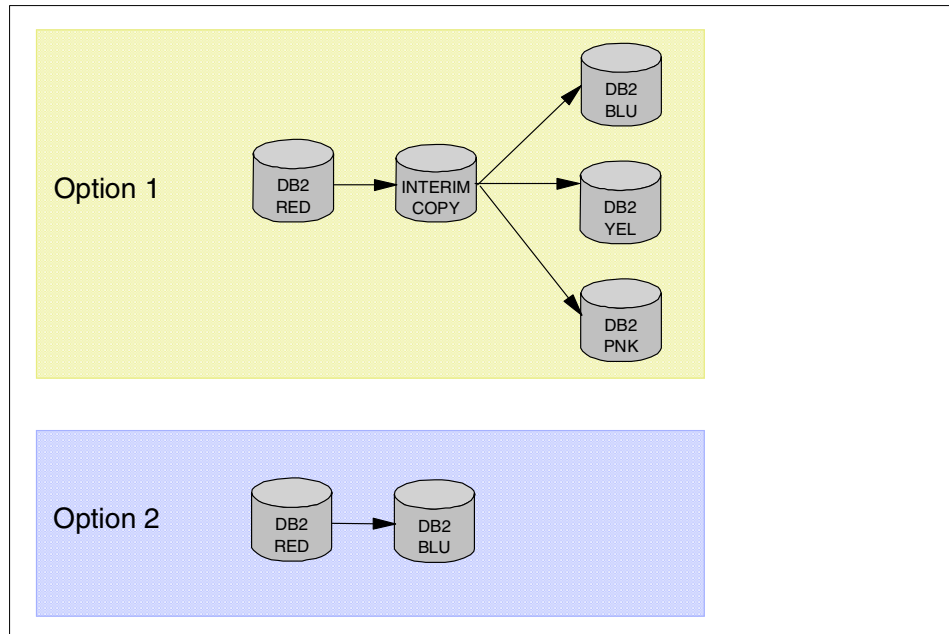


Figure 5-1 HSC using DFDSS copy/rename alternatives

- ▶ Option 1 - Copy source to an intermediate copy
This option is used if multiple target systems are required, or if the source and target cannot be shut down simultaneously.
- ▶ Option 2 - Copy source directly to target
This option is used if only one target system is required, and both source and target can be shut down simultaneously. This is the option we tested and documented in this book.

The configuration we used for Scenario 2 (option 2) is a data sharing to non-data sharing environment. The source system has one active DB2 member. An intermediate LPAR is not necessary to copy and rename the data sets.

Figure 5-2 shows the HSC phases for Scenario 2.

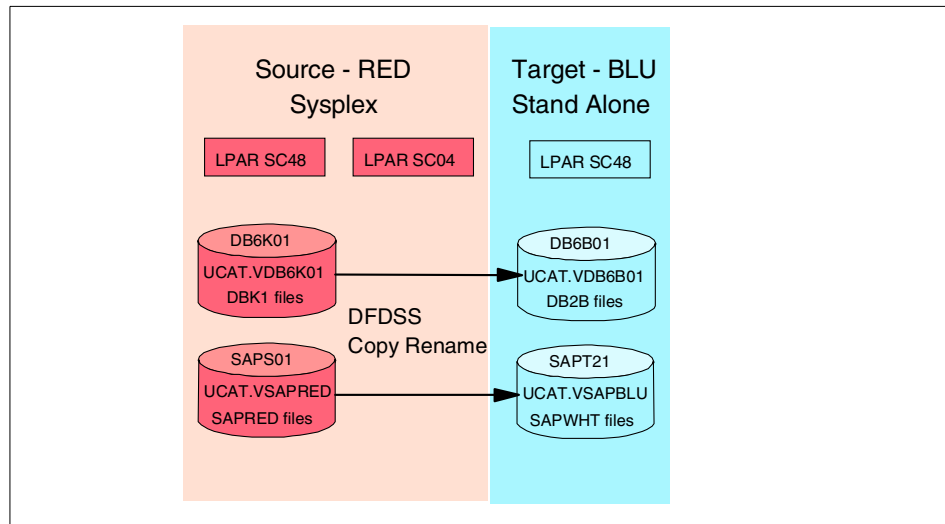


Figure 5-2 Homogeneous system copy using DFDSS copy/rename

This chapter describes the following steps used in our scenario:

1. Preparing the target disk environment
2. Checking the source environment
3. Copying the source system
4. Restarting the target DB2 system
5. Restarting the target SAP system
6. Verification and backup

5.2 Preparing the target disk environment

This section describes the steps that need to be completed prior to replacing the target volumes in the copy/rename step. Back up the environment before completing these steps; once done, the target system will be destroyed.

Execute the following steps to reduce the downtime of your source system:

1. Make sure that all SAP application servers are down on the target system (SAPBLU).
2. Shut down the ICL server and then stop DB2:

```
C ICLIBLU
```

- =DB2B STOP DB2
- 3. Next, delete your SAPBLU user catalog. See Example B-3.
- 4. Ensure that your SAP volumes are offline:
 - S SAPBLU0F
- 5. Reclip the volumes in the SAP SMS pool {Example B-4}.
- 6. Vary your SAP volumes online:
 - S SAPBLU0N
- 7. Recreate the UCAT for the target SAP VSAM data sets {Example B-5}.
- 8. Delete all the VSAM data sets for the DB2 directory, catalog, temp work tablespaces {Example B-6}.

5.3 Checking the source environment

This section describes what you need to check on your source system before starting the offline HSC.

- 1. Ensure that all SAP application servers are down on the source system (SAPRED). Shut down your ICLI server:
 - C ICLIRED
- 2. Shut down the source DB2 system and any associated members:
 - =DBK2 STOP DB2, and
 - =DBK1 STOP DB2
- 3. Start the first source DB2 member:
 - =DBK1 START DB2 ACCESS(MAINT)
- 4. Ensure that no utilities are running or registered in SYSUTILX. Make sure no pagesets are in a restricted status:
 - DISPLAY UTIL(*)
 - DISPLAY DATABASES(*) SPACE(*) LIMIT(*) RESTRICT
- 5. This step can run for a few minutes because of the large number of SAP objects. Shut down the source DB2 system:
 - =DBK1 STOP DB2
- 6. Print BSDS for the source data sharing group {Example B-7}. The purpose for this BSDS print is to provide information for the steps documented in the DB2 system restart phase.

5.4 Copying the source system

At this point we are ready to execute DFDSS copy/rename process of our source system to our target system. This phase executes 22 jobs. See Example A-28 and modify SAPWHT to SAPBLU and DB2V61H1 to DB2V610B1.

Next, ensure that all necessary data sets are copied. Go to ISPF 3.4 and count the data sets with high-level qualifier SAPBLU. The number must be the same as for SAPRED as shown below. Repeat this for DB2 system data sets as well.

Example 5-1 ISPF 3.4 count

```
Total sum of SAPRED is 26833
Total sum of SAPBLU is 26833
```

```
DB2V61K1.DSNDB%.DSNDB06.**
DB2V61K1.DSNDB%.DSNDB01.**
DB2V61K1.DSNDB%.DSNRGFDB.**
DB2V61K1.DSNDB%.DSNRLST.**
DB2V61K1.DSNDB%.DSNDPSM.**
Total Sum of DB2V61K1 is 131
```

```
DB2V610B.DSNDB%.DSNDB06.**
DB2V610B.DSNDB%.DSNDB01.**
DB2V610B.DSNDB%.DSNRGFDB.**
DB2V610B.DSNDB%.DSNRLST.**
DB2V610B.DSNDB%.DSNDPSM.**
Total Sum of DB2V610B is 131
```

5.5 Restarting target DB2 system

The processes required to restart the target DB2 system, as well as what to be aware of, are discussed in the following paragraphs. Keep in mind that the source system is data sharing and the target system is non-data sharing. Thus, a *cold start* needs to be created. This is done by using the source LRSN as RBA on the restart record in the target BSDS.

Unlike with the *online copy*, the target system's original BSDS and logs are used. This is possible because the DB2 is *cold started*, as follows:

1. The LRSN from the source system is extracted because it is required for the restart of the target system. Because the source is a data sharing group, the highest LRSN from all members is required. As all the members of the group are shut down for the copy/rename step, the last checkpoint from each

member is used for the comparison. Check the printout from 5.3, "Checking the source environment" on page 58{Example B-7}.

2. Use the LRSN as the RBA value for the change log inventory job (DSNJU003). The value is the highest LRSN, from the preceding step, with the last three values being set to 0 and the fourth last value being increased by one. Thus, if the last 4 digits are '3F5E', the value would change to '4000'. The RBA for the cold start is the full 12 digits, with the last 4 digits modified as described. To achieve the cold start, the STARTRBA and ENDRBA values must be the same {Example B-8}.
3. Update the ZPARMs to specify DEFER ALL and ensure that the user ID doing the restart has INSTALL SYSADM authority. This is required for the restart process and can be revoked once the system has been restarted. The parameter OFFLOAD is inserted under macro DSN6LOGP in the ZPARMs and set to NO. This is to disable LOG offloads during the following steps.
4. *Ensure IRLM storage is sufficient for the locks used by the alter tablespace/indexes procedures.* Start the target DB2 in maintenance mode for the duration of the restart phase:.

```
=DB2B START DB2 ACCESS(MAINT)
```

Due to the cold start option, DB2 issues a prompt to check that the system is required to restart in the specified mode. Check that the RBA in the prompt is correct as per the preceding step.

5. As the source system is data sharing while the target system is non-data sharing, the work database names could be different, and will be different if the source system has more than one member in the group. These work databases are removed and DSNDB07 is recreated. *Also, check that the same bufferpools are defined on the target system as are on the source system.* This is to ensure that the following ALTERs do not fail with stop pending STOPP on the databases {Example B-9}.

Recreate the work databases as DSNDB07. The SAP indexes created on DSNDB06 are stopped and ALTERed at the same time {Example B-10}.

6. Extract all the STOGROUPs that are to be modified to point to the correct high-level qualifier. The sequence is as follows:
 - Extract all STOGROUPs.
 - Extract all TABLESPACES and INDEXes defined to the preceding STOGROUPs.
 - Stop all the related DATABASEs.
 - ALTER all the related TABLESPACES and INDEXes to a temporary STOGROUP.
 - DROP and reCREATE the STOGROUPs with the new high-level qualifier.

- ALTER all the related TABLESPACES and INDEXes back to the original STOGROUPs.

The information is extracted using REXX procedures. Two indexes are created for the extraction and removed before the ALTER DDL is executed {Example B-11}.

Important: The above steps are recommended by IBM to alter the VCAT name for the STOGROUPS. There are other methods of altering the VCAT name, but you should use them with caution as they are not supported by IBM.

7. Execute the stop of the STOGROUPs and ALTERs of the related DB2 objects. The STOP DATABASE(*) SPACENAM(*) command is used to stop all databases except the DB2 catalog directory databases {Example B-12}
8. On completion of the ALTERs, the target DB2 is checked to ensure that no DB2 objects still reflect the source system VCATs. This is both the DB2 system and SAP high-level qualifiers {Example B-13}. A sample is shown in Example 5-2.

Example 5-2 SELECT to verify VCAT names

```
SELECT  NAME,VCATNAME
        FROM SYSIBM.SYSSTOGROUP
        WHERE VCATNAME = 'DB2V61K1'
        OR    VCATNAME = 'SAPRED' ;
SELECT  TSNAME,STORNAME,VCATNAME,DBNAME
        FROM SYSIBM.SYSTABLEPART
        WHERE VCATNAME = 'DB2V61K1'
        OR    VCATNAME = 'SAPRED' ;
SELECT  IXNAME,STORNAME,VCATNAME
        FROM SYSIBM.SYSINDEXPART
        WHERE VCATNAME = 'DB2V61K1'
        OR    VCATNAME = 'SAPRED' ;
```

9. The target DB2 system is now at the stage where it can be restarted under the correct ZPARM parameters and out of maintenance mode. Prior to restarting the DB2 system, the ZPARMs need to be reset to the original values. Put DEFER ALL back to RESTART ALL, and OFFLOAD NO to OFFLOAD YES.
10. Stop and restart DB2:
=DB2B STOP DB2
=DB2B START DB2

11. The SAP post copy steps can now be executed. Ensure that sensitive SAP data is protected as required and that the correct user IDs have access to the target DB2 system.

5.6 Restarting the target SAP system

Before starting the SAP system, you must install the SAP license. If the target SAP system message server is running on the same hardware as the original install, then you can use the same SAP license that you obtained originally from SAP. If not, you have to obtain a new SAP license from SAP.

Install the SAP license by logging on to the message server, issuing the following command, and supplying the necessary information at the prompts:

```
saplicense -install
```

You should also delete the license for the source system that was copied during the homogeneous system copy process. You do this by entering the following command and supplying the necessary information at the prompts:

```
saplicense -delete
```

Also, before starting the SAP system, ensure that DB2 dual logging and archiving are enabled. In our scenarios, we turned archiving off while we were altering the DB2 catalog. After we completed altering the DB2 catalog, we stopped DB2, reassembled DSNZPARM, and restarted DB2 in normal mode.

Prior to starting the SAP system, review the chapter on General Subsequent Actions of the *SAP R/3 Homogeneous System Copy, Release 4.6C SR2* manual. This chapter documents actions to be taken using native DB2 processes like deleting all rows from a list of SAP tables. It further documents actions to be taken using SAP R/3 transactions like configuring the Workbench Organizer.

If the target system has an SAP application server that is at a different release or executes on a different platform, do the following:

- ▶ Bind packages and plan to be used by the SAP appserver.
- ▶ Perform grants for use of this plan and additional privileges for sidADM.


5.7 Verifications and backup

It is necessary to verify the whole system. At various points in the process, validations were run to make sure that there are no references back to source system VSAM data sets or configuration. We recommend that each installation develop their own SAP application and program product verifications.

These verifications should take into consideration the reason for the homogeneous copy. For instance, it may have been done to bring an SAP application configuration change into the target system. This should be verified. Maybe the source system had business data in it that needed to be used in a certain test process. Validate that the data is as you need it.

Once the restart of the target SAP system is complete, the SAP/DB2 environment must be backed up. The image copy job for the DB2 system objects runs quickly. However, to image copy the SAP objects can take several hours. In order to turn the target system over to the end users quickly, run the image copy job for the DB2 system objects, and if hardware-assisted volume level copy is available, create a copy of the target DB2 system. This will become the first recovery point for the newly created target system.

Get current image copies of the SAP objects after returning the target system to the end users. This can be done via the normal job schedule for image copy jobs.



JCL for Scenario 1 - FlashCopy with suspended DB2

This appendix contains JCL that is more extensive than the JCL included in illustrations in individual chapters. It also includes supplementary JCL we used in steps that were not part of the cloning process.

Example: A-1 SOUSAP01

```
//SAPRES1A JOB (999,POK), 'BUILD JOBS', CLASS=A, MSGCLASS=X,  
//          NOTIFY=&SYSUID, REGION=4M  
/*JOBPARM SYSAFF=SC67  
//DEFINE EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
EXPORT 'UCAT.VSAPWHT' DISCONNECT -  
        CATALOG('MCAT.OS390R10.VO10CAT')
```

Example: A-2 SOUSAP02

```
//SAPRES1A JOB (999,P0K), 'PRINT MAP ', CLASS=A, MSGCLASS=X,  
// NOTIFY=&SYSUID, TIME=1440, REGION=OM  
/*JOBPARM L=9999, SYSAFF=SC48  
//BSDSPRNT EXEC PGM=DSNJU004  
//STEPLIB DD DSN=DSN610.SDSNLOAD, DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DSN=DB2V61K1.DBK1.BSDS01, DISP=SHR  
//SYSIN DD DUMMY  
//
```

Example: A-3 SOUSAP03

```
//SAPRES3A JOB (T,0076), SAPRES3, CLASS=A, MSGCLASS=X,  
// NOTIFY=&SYSUID, REGION=4M  
//* DB2 FLASHCOPY  
//FLASHCOP EXEC PGM=ADRDSU  
//SYSPRINT DD SYSOUT=X  
//SYSUDUMP DD SYSOUT=X, OUTLIM=3000  
//SYSIN DD *  
PARALLEL  
COPY FULL INDYNAM (DB6K01) OUTDYNAM (DB6K07) COPYVOLID ADMIN  
COPY FULL INDYNAM (DB6K02) OUTDYNAM (DB6K08) COPYVOLID ADMIN  
COPY FULL INDYNAM (DB6K03) OUTDYNAM (DB6K09) COPYVOLID ADMIN  
COPY FULL INDYNAM (DB6K04) OUTDYNAM (DB6K10) COPYVOLID ADMIN  
COPY FULL INDYNAM (DB6K05) OUTDYNAM (DB6K11) COPYVOLID ADMIN  
COPY FULL INDYNAM (DB6K06) OUTDYNAM (DB6K12) COPYVOLID ADMIN  
COPY FULL INDYNAM (SAPS01) OUTDYNAM (SAPS11) COPYVOLID ADMIN  
COPY FULL INDYNAM (SAPS02) OUTDYNAM (SAPS12) COPYVOLID ADMIN  
COPY FULL INDYNAM (SAPS03) OUTDYNAM (SAPS13) COPYVOLID ADMIN  
COPY FULL INDYNAM (SAPS04) OUTDYNAM (SAPS14) COPYVOLID ADMIN  
COPY FULL INDYNAM (SAPS05) OUTDYNAM (SAPS15) COPYVOLID ADMIN  
COPY FULL INDYNAM (SAPS06) OUTDYNAM (SAPS16) COPYVOLID ADMIN  
COPY FULL INDYNAM (SAPS07) OUTDYNAM (SAPS17) COPYVOLID ADMIN  
COPY FULL INDYNAM (SAPS08) OUTDYNAM (SAPS18) COPYVOLID ADMIN  
COPY FULL INDYNAM (SAPS09) OUTDYNAM (SAPS19) COPYVOLID ADMIN  
COPY FULL INDYNAM (SAPS10) OUTDYNAM (SAPS20) COPYVOLID ADMIN
```

Example: A-4 RENSAP01

```
//SAPRES1A JOB (999,P0K), 'BUILD JOBS', CLASS=A, MSGCLASS=X,  
// NOTIFY=&SYSUID, REGION=4M  
/*JOBPARM SYSAFF=SC59  
//*-----  
//*-----
```

```

/* UCAT DEFINIEREN UCAT.XXXX * DB2 *
/*
//S0104 EXEC PGM=IDCAMS,REGION=4096K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
EXPORT 'UCAT.VSAPRED' DISCONNECT -
CATALOG('MCAT.OS390R10.VO10CAT')
/*
//S0204 EXEC PGM=IDCAMS,REGION=4096K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
EXPORT 'UCAT.VSAPWHT' DISCONNECT -
CATALOG('MCAT.OS390R10.VO10CAT')
/*
//S0304 EXEC PGM=IDCAMS,REGION=4096K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
EXPORT 'UCAT.VDB6K01' DISCONNECT -
CATALOG('MCAT.OS390R10.VO10CAT')
/*
//

```

Example: A-5 RENSAP02

```

//SAPRES3A JOB (999,P0K),'INIIALIZE VOLUME',MSGCLASS=X,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC48
//INIT1 EXEC PGM=ICKDSF,PARM='NOREPLYU'
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INIT UNIT(600F) VOLID(SAPS11) VFY(SAPS01) -
PURGE VTOC(1,0,400) SG
INIT UNIT(6010) VOLID(SAPS12) VFY(SAPS02) -
PURGE VTOC(1,0,400) SG
INIT UNIT(6019) VOLID(SAPS13) VFY(SAPS03) -
PURGE VTOC(1,0,400) SG
INIT UNIT(601A) VOLID(SAPS14) VFY(SAPS04) -
PURGE VTOC(1,0,400) SG
INIT UNIT(6020) VOLID(SAPS15) VFY(SAPS05) -
PURGE VTOC(1,0,400) SG
INIT UNIT(610C) VOLID(SAPS16) VFY(SAPS06) -
PURGE VTOC(1,0,400) SG
INIT UNIT(610E) VOLID(SAPS17) VFY(SAPS07) -
PURGE VTOC(1,0,400) SG
INIT UNIT(610F) VOLID(SAPS18) VFY(SAPS08) -
PURGE VTOC(1,0,400) SG
INIT UNIT(6119) VOLID(SAPS19) VFY(SAPS09) -
PURGE VTOC(1,0,400) SG

```

```

INIT UNIT(611A) VOLID(SAPS20) VFY(SAPS10) -
    PURGE VTOC(1,0,400) SG
INIT UNIT(6413) VOLID(DB6K07) VFY(DB6K01) -
    PURGE VTOC(1,0,400) SG
INIT UNIT(641F) VOLID(DB6K08) VFY(DB6K02) -
    PURGE VTOC(1,0,400) SG
INIT UNIT(6420) VOLID(DB6K09) VFY(DB6K03) -
    PURGE VTOC(1,0,400) SG
INIT UNIT(6421) VOLID(DB6K10) VFY(DB6K04) -
    PURGE VTOC(1,0,400) SG
INIT UNIT(6422) VOLID(DB6K11) VFY(DB6K05) -
    PURGE VTOC(1,0,400) SG
INIT UNIT(6423) VOLID(DB6K12) VFY(DB6K06) -
    PURGE VTOC(1,0,400) SG
/*

```

Example: A-6 RENSAP03

```

//SAPRES3A JOB (999,P0K), 'BUILD JOBS',CLASS=A,MSGCLASS=X,
//      NOTIFY=&SYSUID,REGION=4M
/*JOBPARM SYSAFF=SC59
/*-----
//STEP01 EXEC PGM=IDCAMS
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
IMPORT CONNECT -
    OBJECTS((UCAT.VSAPRED -
            DEVICETYPE(3390) -
            VOLUMES(SAPS07))) -
    CATALOG(MCAT.VOPICAT)
/*
//STEP02 EXEC PGM=IDCAMS,COND=(4,LT)
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE ALIAS -
    (NAME(SAPRED) -
     RELATE(UCAT.VSAPRED))
/*
//STEP03 EXEC PGM=IDCAMS
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
IMPORT CONNECT -
    OBJECTS((UCAT.VDB6K01 -
            DEVICETYPE(3390) -

```



```

        VOLUMES(DB6K01)) -
        CATALOG(MCAT.VOP1CAT)
/*
//STEP04 EXEC PGM=IDCAMS,COND=(4,LT)
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        DEFINE ALIAS -
        (NAME(DB2V61K1) -
        RELATE(UCAT.VDB6K01))
/*
//

```

Example: A-7 RENSAP04

```

//SAPRES3A JOB (999,POK),'SAPWHT',CLASS=A,MSGLEVEL=(1,1),
// MSGCLASS=X,NOTIFY=&SYSUID
/*JOBPARM L=9999,SYSAFF=SC48
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        DEFINE -
        USERCATALOG -
        (NAME (UCAT.VSAPWHT) -
        ICFCATALOG -
        VOLUMES(SAPT07) -
        STORCLAS(SCSAPWHT) -
        STRNO(8) -
        CYLINDERS(100 50 ) -
        SHAREOPTIONS(3 4)) -
        DATA(BUFND(8) -
        NAME (UCAT.VSAPWHT.DATA) -
        CYLINDERS(100 50) -
        CONTROLINTERVALSIZE(4096) -
        FREESPACE(20 20) -
        RECORDSIZE(100 32400)) -
        INDEX(BUFNI(7) -
        NAME (UCAT.VSAPWHT.INDEX) -
        TRACKS(200,50) -
        CONTROLINTERVALSIZE(4096) -
        IMBED -
        REPLICATE)
/*
/*-----
//S0202 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=X
//SYSIN DD *

```

```
DEF ALIAS(NAME(SAPWHT ) RELATE(UCAT.VSAPWHT          )) -  
  CATALOG(MCAT.VOP1CAT)  
/*  
//
```

Example: A-8 TARSAP01

```
//SAPRES1A JOB (999,P0K),'INIIALIZE VOLUME',MSGCLASS=X,NOTIFY=&SYSUID  
/*JOBPARM SYSAFF=SC67  
//INIT1 EXEC PGM=ICKDSF,PARM='NOREPLYU'  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
  INIT UNIT(6200) VOLID(SAPT01) VFY(SAPT01) -  
    PURGE VTOC(1,0,400) INDEX(28,0,50) SG  
/*  
//VVDS1 EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
  DEFINE CLUSTER -  
    (NAME(SYS1.VVDS.VSAPT01) -  
    TRACKS(180 10) -  
    VOLUMES(SAPT01) -  
    NONINDEXED)  
/*
```

Example: A-9 TARSAP02

```
//SAPRES1A JOB (999,P0K),'PRINT MAP ',CLASS=A,MSGCLASS=X,  
// NOTIFY=&SYSUID,TIME=1440,REGION=OM  
/*JOBPARM L=9999,SYSAFF=SC48  
//BSDSPRNT EXEC PGM=DSNJU004  
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DSN=DB2V61K1.DBK1.BSDS01,DISP=SHR  
//SYSIN DD DUMMY  
//
```

Example: A-10 TARSAP03

```
//SAPRES1A JOB (999,P0K),'PRINT MAP ',CLASS=A,MSGCLASS=X,  
// NOTIFY=&SYSUID,TIME=1440,REGION=OM  
/*JOBPARM L=9999,SYSAFF=SC67  
/*
```

```

/* ----- JCL MODEL -----
/* THIS JOB DROPS SAP INDEXES THAT ARE NOT PART
/* OF THE NORMAL DB2 CATALOG DROP PROCESS.
/*
/* CLOI - ORIGINATING MEMBER SUBSYSTEM ID
/* -----
//IDDELETE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE DB2V61H1.DSNDBC.DSNDB06.SYSTABLE.I0001.A001 CLUSTER
DELETE DB2V61H1.DSNDBC.DSNDB06.SYSTBLSP.I0001.A001 CLUSTER

```

Example: A-11 TARSAP04

```

//SAPRES1A JOB (999,P0K),'DROP DB ',CLASS=A,MSGCLASS=X,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
/*JOBPARM L=9999,SYSAFF=SC67
/*
//JOBLIB DD DISP=SHR,
// DSN=DSN610.SDSNLOAD
//DSNTICU EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DBH1)
-STOP DDF
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) PARM('RCO') -
LIBRARY('DB2V61H1.RUNLIB.LOAD')
END
//SYSIN DD *
DROP DATABASE DSNRGFDB;
DROP DATABASE DSNRLST;
DROP STOGROUP SYSDEFLT;
DROP PROCEDURE SYSPROC.DSNWZP RESTRICT;
DROP PROCEDURE SYSPROC.DSNUTILS RESTRICT;
DROP TABLE SYSIBM.SYSPRINT;
DROP INDEX SAPRES4.INDSTOR;
DROP INDEX SAPRES4.TABSTOR;
/*

```

Example: A-12 TARSAP05

```

//SAPRES1A JOB (999,P0K),'PRINT MAP ',CLASS=A,MSGCLASS=X,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M

```

```

/*JOBPARM L=9999,SYSAFF=SC67
/*
//JOBLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
/*
/* STEP 1: DELETE DATA SETS CREATED BY THE DB2 IVP JOBS
//PH00S01 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DELETE 'DB2V61H1.SYSCOPY.DSN8D61A.DSN8S61D'
DELETE 'DB2V61H1.SYSCOPY.DSN8D61A.DSN8S61E'
DELETE 'DB2V61H1.SYSCOPY.DSN8D61A.DSN8S61P'
DELETE 'DB2V61H1.SYSCOPY.DSN8D61A.DSN8S61C'
/*
/* STEP 2: FREE PLANS AND PACKAGES CREATED BY DB2 IVP JOBS
//PH00S02 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DBH1)
FREE PLAN(DSN8BD61)
FREE PLAN(DSN8BH61)
FREE PLAN(DSN8BF61)
FREE PLAN(DSN8BP61)
FREE PLAN(DSN8LC61)
FREE PLAN(DSN8LL61)
FREE PLAN(DSN8LP61)
FREE PLAN(DSN8LR61)
FREE PLAN(DSN8IC0)
FREE PLAN(DSN8IP0)
FREE PLAN(DSN8IQ0)
FREE PLAN(DSN8IH0)
FREE PLAN(DSN8CC0)
FREE PLAN(DSN8CP0)
FREE PLAN(DSN8CQ0)
FREE PLAN(DSN8CH0)
FREE PLAN(DSN8ED1)
FREE PLAN(DSN8EP1)
FREE PLAN(DSN8SC61)
FREE PLAN(DSN8HC61)
FREE PLAN(DSN8SP61)
FREE PLAN(DSN8UW61)
FREE PACKAGE(DSN8CC61.*)
FREE PACKAGE(DSN8ED61.*)
FREE PACKAGE(DSN8EP61.*)
FREE PACKAGE(DSN8HC61.*)
FREE PACKAGE(DSN8IC61.*)

```

```

FREE PACKAGE(DSN8LC61.*)
FREE PACKAGE(DSN8LL61.*)
FREE PACKAGE(DSN8LP61.*)
FREE PACKAGE(DSN8LR61.*)
FREE PACKAGE(DSN8DU61.*)
END
/**
/**      3: DROP TABLES, DB'S AND STOGROUPS CREATED BY DB2 IVP JOBS
//PH00S03 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DBH1)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
    LIB('DB2V61H1.RUNLIB.LOAD')
END
//SYSIN DD *
DROP TABLE DSN861SA.STAFF;
COMMIT;
DROP TABLE DSN861SA.TESTSTUFF;
COMMIT;
DROP DATABASE DSN8D61A;
COMMIT;
DROP DATABASE DSN8D61L;
COMMIT;
DROP DATABASE DSN8D61P;
COMMIT;
DROP DATABASE DSN8D61U;
COMMIT;
DROP STOGROUP DSN8G610;
COMMIT;
DROP STOGROUP DSN8G61U;
COMMIT;
/**
/**      4: DROP USER-DEFINED FUNCTIONS CREATED BY DB2 IVP JOBS
//PH00S04 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DBH1)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
    LIB('DB2V61H1.RUNLIB.LOAD')
END
//SYSIN DD *
DROP SPECIFIC FUNCTION DSN8.DSN8DUCDDVV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUCDVVV RESTRICT;

```

```

COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUADV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUCTTVV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUCTVVV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUATV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUCYFV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUCYFVV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8EUDND RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8EUDNV RESTRICT;
COMMIT;
/*
/*      5: DROP USER-DEFINED FUNCTIONS CREATED BY DB2 IVP JOBS (CONT)
//PHOOS05 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DBH1)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
    LIB('DB2V61H1.RUNLIB.LOAD')
END
//SYSIN DD *
DROP SPECIFIC FUNCTION DSN8.DSN8EUMND RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8EUMNV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUTINV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUTINVV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUTINVVV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUTISV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUTISVV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUTISVVV RESTRICT;
COMMIT;
/*
/*      6: DROP USER-DEFINED FUNCTIONS CREATED BY DB2 IVP JOBS (CONT)
//PHOOS06 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*

```

```

//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DBH1)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
    LIB('DB2V61H1.RUNLIB.LOAD')
END
//SYSIN DD *
DROP SPECIFIC FUNCTION DSN8.DSN8DUTILV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUTILVV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUTILVVV RESTRICT;
COMMIT;
DROP SPECIFIC FUNCTION DSN8.DSN8DUWFV RESTRICT;
COMMIT;
//*

```

Example: A-13 TARSAP06

```

//SAPRES1A JOB (999,P0K), 'PRINT MAP ',CLASS=A,MSGCLASS=X,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
/*JOBPARM L=9999,SYSAFF=SC67
//DSNTIC PROC
/* ***** */
/* DIRECTORY/CATALOG AMS INVOCATION INSTREAM JCL PROCEDURE */
/* ***** */
//DSNTIC EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DSNTIC PEND
//DSNTINV EXEC DSNTIC
//SYSIN DD *
  DELETE (DB2V61H1.RUNLIB.LOAD)

  DELETE (DB2V61H1.SRCLIB.DATA)

  DELETE -
    (DB2V61H1.DSNDBC.DSNDB01.DBD01.I0001.A001) -
    CATALOG(DB2V61H1)
  DELETE -
    (DB2V61H1.DSNDBC.DSNDB01.SCT02.I0001.A001) -
    CATALOG(DB2V61H1)
  DELETE -
    (DB2V61H1.DSNDBC.DSNDB01.DSNSCT02.I0001.A001) -
    CATALOG(DB2V61H1)

```

```

DELETE      -
            (DB2V61H1.DSNDBC.DSNDB01.SPT01.I0001.A001) -
            CATALOG(DB2V61H1)
DELETE      -
            (DB2V61H1.DSNDBC.DSNDB01.DSNSPT01.I0001.A001) -
            CATALOG(DB2V61H1)
DELETE      -
            (DB2V61H1.DSNDBC.DSNDB01.DSNSPT02.I0001.A001) -
            CATALOG(DB2V61H1)
DELETE      -
            (DB2V61H1.DSNDBC.DSNDB01.SYSLGRNX.I0001.A001) -
            CATALOG(DB2V61H1)
And so on for DSNDB06

```

Example: A-14 TARSAP07

```

//SAPRES1A JOB (999,POK), 'PRINT MAP ', CLASS=A, MSGCLASS=X,
// NOTIFY=&SYSUID, TIME=1440, REGION=OM
/*JOBPARM L=9999, SYSAFF=SC67
/* ----- JCL MODEL -----
/* THIS JOB UNCATALOGS DB2 ARCHIVE DATA SETS
/* CLOI - ORIGINATING MEMBER SUBSYSTEM ID
/* CLO/MI - EITHER THE ORIGINATING MEMBER SUBSYSTEM OR
/*          THE NAME OF A MEMBER IN THE DATA SHARING
/*          GROUP.
/* -----
/******
//STEPONE EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1      DD DSN=DB2V61H1.DBH1.ARCLG1.A0000001, DISP=(OLD,DELETE)
//DD2      DD DSN=DB2V61H1.DBH1.ARCLG1.A0000002, DISP=(OLD,DELETE)
//DD3      DD DSN=DB2V61H1.DBH1.ARCLG1.A0000003, DISP=(OLD,DELETE)
//DD4      DD DSN=DB2V61H1.DBH1.ARCLG1.A0000004, DISP=(OLD,DELETE)
//DD5      DD DSN=DB2V61H1.DBH1.ARCLG1.A0000005, DISP=(OLD,DELETE)
//DD6      DD DSN=DB2V61H1.DBH1.ARCLG1.A0000006, DISP=(OLD,DELETE)
//DD7      DD DSN=DB2V61H1.DBH1.ARCLG1.A0000007, DISP=(OLD,DELETE)

```

Example: A-15 TARSAP08

```

//SAPRES1A JOB (999,POK), 'BUILD JOBS', CLASS=A, MSGCLASS=X,
//          NOTIFY=&SYSUID, REGION=4M
/*JOBPARM SYSAFF=SC67
/*
/* RESTORE ARCHIVE LOG COPY TO DISK FROM SOURCE SYSTEM ARCHIVE

```



```

/* TO TARGET SYSTEM ARCHIVE DATASET NAME.
/*
//STEP21 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=SAPRES2.DBK1.ARCLG1.A0000061,
// DISP=SHR
//SYSUT2 DD DSN=SAPRES2.DBH1.ARCLG1.A0000001,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1700,100),RLSE),
// DCB=(LRECL=4096,RECFM=FB,BLKSIZE=28672)
//SYSIN DD DUMMY
/*

```

Example: A-16 TARSAP09

```

//SAPRES1A JOB (999,P0K), 'BUILD JOBS',CLASS=A,MSGCLASS=X,
// NOTIFY=&SYSUID,REGION=4M
/*JOBPARM SYSAFF=SC67
/*ROUTE PRINT LOCAL
/*
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        REPRO IDS(DB2V61H1.DBH1.SKELETON.BSDS01) -
            ODS(DB2V61H1.DBH1.BSDS01) REUSE
        REPRO IDS(DB2V61H1.DBH1.SKELETON.BSDS01) -
            ODS(DB2V61H1.DBH1.BSDS02) REUSE
/*
//

```

Example: A-17 BSDS updates for DB2 restart using active logs

```

//SAPRES1A JOB (999,P0K), 'BUILD JOBS',CLASS=A,MSGCLASS=X,
// NOTIFY=&SYSUID
/*JOBPARM SYSAFF=SC67
/*
/* DELETED AND RE-ADD NEW ACTIVE LOGS
/*
//ACTLOG EXEC PGM=DSNJU003
//STEPLIB DD DISP=SHR,DSN=DSN610.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS01
//SYSUT2 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *

```

```

DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS01
DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS02
DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS03
DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY2.DS01
DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY2.DS02
DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY2.DS03
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS01,COPY1
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS02,COPY1
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS03,COPY1,
STARTRBA=001293173000,ENDRBA=0012A1658DFFF
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS01,COPY2
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS02,COPY2
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS03,COPY2,
STARTRBA=001293173000,ENDRBA=0012A1658DFFF
/**
/**  CHANGE THE HIGHEST RBA WRITTEN
/**
/**HIGHRBA EXEC PGM=DSNJU003
/**STEPLIB DD DISP=SHR,DSN=DSN610.SDSNLOAD
/**SYSUT1 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS01
/**SYSUT2 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS02
/**SYSPRINT DD SYSOUT=*
/**SYSUDUMP DD SYSOUT=*
/**SYSIN DD *
HIGHRBA STARTRBA=001293DC0B77,OFFLRBA=001293DB2FFF,
TIME=20011590604040
/**
/**  ADD THE RESTART RECORD TO THE BSDS - WITHOUT TRUNCATION
/**
/**CRESTART EXEC PGM=DSNJU003
/**STEPLIB DD DISP=SHR,DSN=DSN610.SDSNLOAD
/**SYSUT1 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS01
/**SYSUT2 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS02
/**SYSPRINT DD SYSOUT=*
/**SYSUDUMP DD SYSOUT=*
/**SYSIN DD *
CRESTART CREATE,CHKPTRBA=001293658EB6

```

Example: A-18 TARSAP10

```

//SAPRES1A JOB (999,POK), 'BUILD JOBS',CLASS=A,MSGCLASS=X,
// NOTIFY=&SYSUID,REGION=4M
/**JOBPARM SYSAFF=SC67
/**
/** CHANGE LOG INVENTORY:

```

```

/**      UPDATE BSDS WITH NEW ACTIVE LOGS
/**
//ACTLOG EXEC PGM=DSNJU003
//STEPLIB DD DISP=SHR,DSN=DSN610.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS01
//SYSUT2 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS01
DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS02
DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS03
DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY2.DS01
DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY2.DS02
DELETE DSNAME=DB2V61H1.DBH1.LOGCOPY2.DS03
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS01,COPY1
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS02,COPY1
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY1.DS03,COPY1
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY2.DS01,COPY2
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY2.DS02,COPY2
NEWLOG DSNAME=DB2V61H1.DBH1.LOGCOPY2.DS03,COPY2
/**
//ARCHLOG EXEC PGM=DSNJU003
//STEPLIB DD DISP=SHR,DSN=DSN610.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS01
//SYSUT2 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
NEWLOG DSNAME=SAPRES2.DBH1.ARCLG1.A0000001,COPY1VOL=TOTSTU,
UNIT=3390,STARTRBA=00128A6DE000,ENDRBA=0012B65FDFFF,
STRTRLRSN=B5FB4F6963A0,ENDLRSN=B5FC3A42AFAF,CATALOG=YES
/**
//CHECKPT EXEC PGM=DSNJU003
//STEPLIB DD DISP=SHR,DSN=DSN610.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS01
//SYSUT2 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
CHECKPT STARTRBA=001293658EB6,ENDRBA=001293682A22,
ENDLRSN=B5FC2DF54747,TIME=2001165134838
/**
//CRESTART EXEC PGM=DSNJU003
//STEPLIB DD DISP=SHR,DSN=DSN610.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS01
//SYSUT2 DD DISP=OLD,DSN=DB2V61H1.DBH1.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*

```

```
//SYSIN DD *
CRESTART CREATE, ENDLRSN=B5FC2E8726FF,CHKPTRBA=001293658EB6,
FORWARD=YES,BACKOUT=YES
```

Example: A-19 TARSAP11

```
//SAPRES1A JOB (999,P0K), 'PRINT MAP ', CLASS=A, MSGCLASS=X,
// NOTIFY=&SYSUID, TIME=1440, REGION=OM
/*JOBPARM L=9999, SYSAFF=SC48
//BSDSPRNT EXEC PGM=DSNJU004
//STEPLIB DD DSN=DSN610.SDSNLOAD, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=DB2V61H1.DBH1.BSDS01, DISP=SHR
//SYSIN DD DUMMY
//
```

Example: A-20 TARSAP12

```
//SAPRES1A JOB (999,P0K), 'DBH1 INSTALL', CLASS=A, MSGCLASS=X,
// NOTIFY=&SYSUID, TIME=1440, REGION=OM
/*JOBPARM L=9999, SYSAFF=SC67
// JCLLIB ORDER=(DSN610.PROCLIB)
//DSNTIZA EXEC PGM=ASMA90, PARM='OBJECT,NODECK'
//SYSLIB DD DISP=SHR,
// DSN=DSN610.SDSNMACS
// DD DISP=SHR,
// DSN=SYS1.MACLIB
//SYSLIN DD DSN=&&LOADSET(DSNTILM1), DISP=(NEW,PASS),
// UNIT=SYSALLDA,
// SPACE=(800,(50,50,2)), DCB=(BLKSIZE=800)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSUT1 DD UNIT=SYSALLDA, SPACE=(800,(50,50),,,ROUND)
//SYSUT2 DD UNIT=SYSALLDA, SPACE=(800,(50,50),,,ROUND)
//SYSUT3 DD UNIT=SYSALLDA, SPACE=(800,(50,50),,,ROUND)
//SYSIN DD *
DSN6ENV MVS=XA
DSN6SPRM DEFER, X
ALL, X
ABEXP=YES, X
```

Example: A-21 TARSAP13

```
//SAPRES1A JOB (999,POK),'BUILD JOBS',CLASS=A,MSGCLASS=X,
//          NOTIFY=&SYSUID,REGION=4M
/*JOBPARM SYSAFF=SC67
//*
//*****/
/* D S N T E M P                                     */
//*                                               */
/*          THIS JOB IS USED TO DROP TEMP DATA SETS.  */
//*                                               */
//*****/
//*
//DSNTIST EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(DBH1)
    -STOP DATABASE(DSNDB06) SPACENAM(SYSTBLSP)
    -STOP DATABASE(DSNDB06) SPACENAM(SYSTABLE)
    -STOP DATABASE(DBK1WRK)
    -STOP DATABASE(DBK2WRK)
    RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) PARM('RCO') -
        LIB('DB2V61K1.RUNLIB.LOAD')
    -START DATABASE(DSNDB06) SPACENAM(SYSTBLSP)
    -START DATABASE(DSNDB06) SPACENAM(SYSTABLE)
    END
//SYSIN DD *
    CREATE STOGROUP TEMPSAP VOLUMES("*) VCAT DB2V61H1 ;
    COMMIT ;
    ALTER INDEX SAPR3."SYSTABLE~0" USING STOGROUP TEMPSAP;
    ALTER INDEX SAPR3."SYSTBLSP~0" USING STOGROUP TEMPSAP;
/*
//*
//DSNTIST EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(DBH1)
    RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) PARM('RCO') -
        LIB('DB2V61K1.RUNLIB.LOAD')
    END
//SYSIN DD *
    DROP DATABASE DBK1WRK ;
```

```

DROP DATABASE DBK2WRK ;
/*

```

Example: A-22 TARSAP14

```

//SAPRES1A JOB (999,POK), 'BUILD JOBS', CLASS=A, MSGCLASS=X,
//          NOTIFY=&SYSUID, REGION=4M
//*JOBPARM SYSAFF=SC67
/*
/*****/
/* D S N T E M P                                     */
/*                                                     */
/*          THIS JOB IS USED TO CREATE THE WORK DATABASE AND TABLESPACES */
/*                                                     */
/*****/
//DSNTICR EXEC PGM=IKJEFT01, DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD, DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
        DSN SYSTEM(DBH1)
        RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) PARM('RCO') -
          LIB('DB2V61K1.RUNLIB.LOAD')
        END
//SYSIN DD *
        CREATE DATABASE DBH1WRK AS WORKFILE FOR DBH1 ;
/*
/*
//DSNTTMP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
/*****/
/* DEFINE TEMPORARY TABLESPACES                                     */
/* THESE TABLE SPACES ARE USED FOR TEMPORARY OR INTERMEDIATE      */
/* TABLES BY DB2, IN SORTING FOR ONE EXAMPLE.                       */
/*****/
//SYSIN DD DSN=DB2V61H1.VSAMPARM(DSN4K01), DISP=SHR
//          DD DSN=DB2V61H1.VSAMPARM(DSN4K02), DISP=SHR
//          DD DSN=DB2V61H1.VSAMPARM(DSN4K03), DISP=SHR
//          DD DSN=DB2V61H1.VSAMPARM(DSN4K04), DISP=SHR
//          DD DSN=DB2V61H1.VSAMPARM(DSN32K01), DISP=SHR
//          DD DSN=DB2V61H1.VSAMPARM(DSN32K02), DISP=SHR
/*
//DSNTIST EXEC PGM=IKJEFT01, DYNAMNBR=20,
//          COND=(4,LT,DSNTTMP)
//STEPLIB DD DSN=DSN610.SDSNLOAD, DISP=SHR

```

```

//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(DBH1)
  -STOP DATABASE(DBH1WRK)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
    LIB('DB2V61K1.RUNLIB.LOAD')
  -START DATABASE(DBH1WRK)
  END
//SYSIN DD *
CREATE TABLESPACE DSN4K01 IN DBH1WRK
  BUFFERPOOL BP0
  CLOSE NO
  USING VCAT DB2V61H1;
CREATE TABLESPACE DSN4K02 IN DBH1WRK
  BUFFERPOOL BP0
  CLOSE NO
  USING VCAT DB2V61H1;
CREATE TABLESPACE DSN4K03 IN DBH1WRK
  BUFFERPOOL BP0
  CLOSE NO
  USING VCAT DB2V61H1;
CREATE TABLESPACE DSN4K04 IN DBH1WRK
  BUFFERPOOL BP0
  CLOSE NO
  USING VCAT DB2V61H1;
CREATE TABLESPACE DSN32K01 IN DBH1WRK
  BUFFERPOOL BP32K
  CLOSE NO
  USING VCAT DB2V61H1;
CREATE TABLESPACE DSN32K02 IN DBH1WRK
  BUFFERPOOL BP32K
  CLOSE NO
  USING VCAT DB2V61H1;
/*

```

Example: A-23 TARSAP15

```

//SAPRES1A JOB (999,P0K), 'BUILD JOBS', CLASS=A, MSGCLASS=X,
//          NOTIFY=&SYSUID, REGION=4M
/*JOBPARM SYSAFF=SC67
/*

```

```

//*****
//*
//DSNTIST EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(DBH1)
    RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) PARM('RC0') -
        LIB('DB2V61K1.RUNLIB.LOAD')
    END
//SYSIN DD *
    CREATE INDEX SAPRES4.INDSTOR
        ON SYSIBM.SYSINDEXPART
        (STORNAME)
        USING STOGROUP TEMPSAP
    ;
    CREATE INDEX SAPRES4.TABSTOR
        ON SYSIBM.SYSTABLEPART
        (STORNAME)
        USING STOGROUP TEMPSAP
    ;
/*
/*-----*//
/* REXX TO EXTRACT STOGROUP NAMES FROM THE TARGET SYSTEM AFTER *//
/* DB2 HAS BEEN RESTARTED. *//
/* PARAMETERS: *//
/* ----- *//
/* 1) REXX PROC TO EXECUTE *//
/* 2) TURN TRACING ON / OFF *//
/* 3) DB2 SUB-SYSTEM TO CONNECT TO *//
/* 4) VCATNAME OF THE NEW HLQ *//
/* 5) CREATOR ID OF EXISTING STOGROUP *//
/* 6) CREATOR ID FOR NEW STOGROUP *//
/* *//
/*-----*//
//STEP01 EXEC PGM=IKJEFT01,DYNAMNBR=60,REGION=OK,COND=(4,LT),
// PARM='%DB2REXX1 N DBH1 DB2V61H1 RCONWAY SAPRES1'
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
// DD DSN=DB2V61K1.RUNLIB.LOAD,DISP=SHR
//SYSPROC DD DSN=SAPRES5.JCL.SCENAR1.LIBRARY,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DB21OUT1 DD DSN=SAPRES4.TEMP.SDROPSTO,DISP=(,CATLG,DELETE),
// SPACE=(TRK,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB21OUT2 DD DSN=SAPRES4.TEMP.SCREASTO,DISP=(,CATLG,DELETE),
// SPACE=(TRK,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//SYSIN DD DUMMY

```



```

//SYSTSIN DD DUMMY
//*
//*-----*//
/** REXX TO EXTRACT TABLESPACE / INDEXSPACE TO ALTER PRIOR TO **//
/** CHANING THE VCAT NAME. **//
/** PARAMETERS: **//
/** ----- **//
/** 1) REXX PROC TO EXECUTE **//
/** 2) TURN TRACING ON / OFF **//
/** 3) DB2 SUB-SYSTEM TO CONNECT TO **//
/** 4) VCATNAME OF THE NEW HLQ **//
/** 5) CREATOR ID FOR NEW STOGROUP **//
/** 6) CHECK FOR SAP INDICES ON DSNDB06 **//
/** -----*//
//STEPO2 EXEC PGM=IKJEFT01,DYNAMNBR=60,REGION=0K,COND=(4,LT),
// PARM='%DB2REXX2 N DBH1 DB2V61H1 SAPRES1 Y'
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
// DD DSN=DB2V61K1.RUNLIB.LOAD,DISP=SHR
//SYSPROC DD DSN=SAPRES5.JCL.SCENAR1.LIBRARY,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DB21INP1 DD DSN=SAPRES4.TEMP.SDROPSTO,DISP=SHR
/**
//DB22OUT1 DD DSN=SAPRES4.TEMP.SALTPERM,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT2 DD DSN=SAPRES4.TEMP.SALTTEMP,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT3 DD DSN=SAPRES4.TEMP.SSTOPDB,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT4 DD DSN=SAPRES4.TEMP.SSTRADB,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//SYSIN DD DUMMY
//SYSTSIN DD DUMMY
/**
/**-----*//
/** REXX TO EXTRACT STOGROUP NAMES FROM THE TARGET SYSTEM AFTER **//
/** DB2 HAS BEEN RESTARTED. **//
/** PARAMETERS: **//
/** ----- **//
/** 1) REXX PROC TO EXECUTE **//
/** 2) TURN TRACING ON / OFF **//
/** 3) DB2 SUB-SYSTEM TO CONNECT TO **//
/** 4) VCATNAME OF THE NEW HLQ **//
/** 5) CREATOR ID OF EXISTING STOGROUP **//
/** 6) CREATOR ID FOR NEW STOGROUP **//
/** -----*//
//STEP03 EXEC PGM=IKJEFT01,DYNAMNBR=60,REGION=0K,

```

```

// PARM='%DB2REXX1 N DBH1 SAPWHT SAPR3 SAPR3'
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
// DD DSN=DB2V61K1.RUNLIB.LOAD,DISP=SHR
//SYSPROC DD DSN=SAPRES5.JCL.SCENAR1.LIBRARY,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DB21OUT1 DD DSN=SAPRES4.TEMP.DROPSTO,DISP=(,CATLG,DELETE),
// SPACE=(TRK,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB21OUT2 DD DSN=SAPRES4.TEMP.CREASTO,DISP=(,CATLG,DELETE),
// SPACE=(TRK,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//SYSIN DD DUMMY
//SYSTSIN DD DUMMY
//*
//*-----*//
//* REXX TO EXTRACT TABLESPACE / INDEXSPACE TO ALTER PRIOR TO **//
//* CHANING THE VCAT NAME. **//
//* PARAMETERS: **//
//* ----- **//
//* 1) REXX PROC TO EXECUTE **//
//* 2) TURN TRACING ON / OFF **//
//* 3) DB2 SUB-SYSTEM TO CONNECT TO **//
//* 4) VCATNAME OF THE NEW HLQ **//
//* 5) CREATOR ID FOR NEW STOGROUP **//
//* 6) CHECK FOR SAP INDICES ON DSNDB06 **//
//* **//
//*-----*//
//STEP04 EXEC PGM=IKJEFT01,DYNAMNBR=60,REGION=OK,COND=(4,LT),
// PARM='%DB2REXX2 N DBH1 SAPWHT SAPR3 N'
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
// DD DSN=DB2V61K1.RUNLIB.LOAD,DISP=SHR
//SYSPROC DD DSN=SAPRES5.JCL.SCENAR1.LIBRARY,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DB21INP1 DD DSN=SAPRES4.TEMP.DROPSTO,DISP=SHR
//*
//DB22OUT1 DD DSN=SAPRES4.TEMP.ALTPERM,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT2 DD DSN=SAPRES4.TEMP.ALTTMP,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT3 DD DSN=SAPRES4.TEMP.STOPDB,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT4 DD DSN=SAPRES4.TEMP.STRTDB,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//SYSIN DD DUMMY
//SYSTSIN DD DUMMY
//*
//*
//DSNTIST EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR

```

```

//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(DBH1)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) PARM('RCO') -
    LIB('DB2V61K1.RUNLIB.LOAD')
  END
//SYSIN DD *
  DROP INDEX SAPRES4.INDSTOR ;
  DROP INDEX SAPRES4.TABSTOR ;
/*
//

```

Example: A-24 TARSAP16

```

//SAPRES1A JOB (999,POK), 'BUILD JOBS', CLASS=A, MSGCLASS=X,
//          NOTIFY=&SYSUID, REGION=4M
/*JOBPARM SYSAFF=SC67
//*
//*****
//DSNTIST EXEC PGM=IKJEFT01, DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD, DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(DBH1)
  -STOP DATABASE(DSNDB06) SPACENAM(SYSTBLSP)
  -STOP DATABASE(DSNDB06) SPACENAM(SYSTABLE)
//          DD DSN=SAPRES4.TEMP.SSTOPDB, DISP=SHR
//          DD *
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
    LIB('DB2V61K1.RUNLIB.LOAD')
  -START DATABASE(DSNDB06) SPACENAM(SYSTBLSP)
  -START DATABASE(DSNDB06) SPACENAM(SYSTABLE)
  END
//*
//SYSIN DD DSN=SAPRES4.TEMP.SALTTEMP, DISP=SHR
//          DD DSN=SAPRES4.TEMP.SDROPSTO, DISP=SHR
//          DD DSN=SAPRES4.TEMP.SCREASTO, DISP=SHR
//          DD DSN=SAPRES4.TEMP.SALTPERM, DISP=SHR
//          DD *
  ALTER INDEX SAPR3."SYSTABLE~0" USING STOGROUP SYSDEFLT ;
  ALTER INDEX SAPR3."SYSTBLSP~0" USING STOGROUP SYSDEFLT ;
  COMMIT;
  DROP STOGROUP TEMPSAP ;

```

```

/**
//DSNTIST EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(9,LT)
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(DBH1)
    RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
        LIB('DB2V61K1.RUNLIB.LOAD')
//      DD DSN=SAPRES4.TEMP.STRTDB,DISP=SHR
//      DD *
    END
/**
//SYSIN DD DSN=SAPRES4.TEMP.ALTTEMP,DISP=SHR
//      DD DSN=SAPRES4.TEMP.DROPSTO,DISP=SHR
//      DD DSN=SAPRES4.TEMP.CREASTO,DISP=SHR
//      DD DSN=SAPRES4.TEMP.ALTPERM,DISP=SHR
/**

```

Example: A-25 TARSAP17

```

//SAPRES1A JOB (999,POK), 'BUILD JOBS',CLASS=A,MSGCLASS=X,
//      NOTIFY=&SYSUID,REGION=4M
/*JOBPARM SYSAFF=SC67
/**
//DSNTIST EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(DBH1)
    RUN PROGRAM(DSNTPE2) PLAN(DSNTPE61) -
        LIB('DB2V61K1.RUNLIB.LOAD')
    END
/**
//SYSIN DD *
SELECT  NAME,VCATNAME
        FROM SYSIBM.SYSSTOGROUP
        WHERE VCATNAME = 'DB2V61K1' OR
              VCATNAME = 'SAPRED'      ;
SELECT  TSNAME,STORNAME,VCATNAME
        FROM SYSIBM.SYSTABLEPART
        WHERE VCATNAME = 'DB2V61K1' OR
              VCATNAME = 'SAPRED'      ;
SELECT  IXNAME,STORNAME,VCATNAME

```

```

FROM SYSIBM.SYSINDEXPART
WHERE VCATNAME = 'DB2V61K1' OR
      VCATNAME = 'SAPRED' ;
/*****

```

Example: A-26 TARSAP18

```

//SAPRES1A JOB (999,P0K), 'BUILD JOBS', CLASS=A, MSGCLASS=X,
//          NOTIFY=&SYSUID, REGION=4M
/*JOBPARM SYSAFF=SC67
/*-----
//STEP01 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
EXPORT 'UCAT.VSAPWHT' DISCONNECT          -
      CATALOG('MCAT.OS390R10.VO10CAT')
/*
//STEP02 EXEC PGM=IDCAMS
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
IMPORT CONNECT -
      OBJECTS((UCAT.VSAPWHT -
              DEVICETYPE(3390) -
              VOLUMES(SAPT08))) -
      CATALOG(MCAT.OS390R10.VO10CAT)
/*
//STEP03 EXEC PGM=IDCAMS, COND=(4,LT)
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
DEFINE ALIAS -
      (NAME(SAPWHT) -
      RELATE(UCAT.VSAPWHT))
/*

```

Example: A-27 TARSAP19

```

//SAPRES1A JOB (999,P0K), 'DBH1 INSTALL', CLASS=A, MSGCLASS=X,
//          NOTIFY=&SYSUID, TIME=1440, REGION=0M
/*JOBPARM L=9999, SYSAFF=SC67
// JCLLIB ORDER=(DSN610.PROCLIB)
//DSNTIZA EXEC PGM=ASMA90, PARM='OBJECT,NODECK'
//SYSLIB  DD DISP=SHR,

```

```

//          DSN=DSN610.SDSNMACS
//          DD  DISP=SHR,
//          DSN=SYS1.MACLIB
//SYSLIN   DD  DSN=&&LOADSET(DSNTILM1),DISP=(NEW,PASS),
//          UNIT=SYSALLDA,
//          SPACE=(800,(50,50,2)),DCB=(BLKSIZE=800)
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSALLDA,SPACE=(800,(50,50),,,ROUND)
//SYSUT2   DD  UNIT=SYSALLDA,SPACE=(800,(50,50),,,ROUND)
//SYSUT3   DD  UNIT=SYSALLDA,SPACE=(800,(50,50),,,ROUND)
//SYSIN    DD  *
          DSN6ENV   MVS=XA
          DSN6SPRM  RESTART,
                   ALL,
                   ABEXP=YES,

```

```

X
X
X

```

Example: A-28 RENAMDB2

```

//SAPRES1A JOB (T,0076),SAPRES1,CLASS=A,MSGCLASS=X,
//          NOTIFY=&SYSUID,REGION=4M
//*          DB2 FLASHCOPY
//COPYRENA EXEC PGM=ADRDSU
//SYSPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X,OUTLIM=3000
//SYSIN    DD *
          COPY DATASET(INCLUDE(DB2V61K1.DSNDB%.DSNDB06.** -
                                DB2V61K1.DSNDB%.DSNDB01.** -
                                DB2V61K1.DSNDB%.DSNRGFDB.** -
                                DB2V61K1.DSNDB%.DSNRLST.** -
                                DB2V61K1.DSNDB%.DSNDPSM.**)) -
          RENAMEU((DB2V61K1.**,DB2V61H1.**)) -
          ALLDATA(*) ALLX TOL(ENQF) -
          NULLMGMTCLAS -
          CATALOG
/*
//*          DB2 FLASHCOPY
//FLASHCOP EXEC PGM=ADRDSU
//SYSPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X,OUTLIM=3000
//SYSIN    DD *
          COPY DATASET(INCLUDE(SAPRED.DSNDB%.A201*.**)) -
          RENAMEU((SAPRED.**,SAPWHT.**)) -
          ALLDATA(*) ALLX TOL(ENQF) -
          NULLMGMTCLAS -
          CATALOG

```

```

/*
/**
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.A202*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.A203*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.A200*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.A0*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.A1*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.A21*.* -
                        SAPRED.DSNDB%.A22*.* -
                        SAPRED.DSNDB%.A23*.* -
                        SAPRED.DSNDB%.A24*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.U0*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.U1*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.CL*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.PO*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.DI*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.DO*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.LO*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.PR*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.LD*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.LX*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.SD*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.SX*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.SO*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.DSNDB%.TM*.*)) -
/** COPY DATASET(INCLUDE(SAPRED.**))EXCLUDE(SAPRED.DSNDB%.A202*.*.* -
                        ..... -
                        SAPRED.DSNDB%.TM*.*)) -

```

Example: A-29 ESA.SYS1.PROCLIB(SAPWHTON)

```

//SAPWHTON PROC
/**
// COMMAND 'RO *ALL,V 6200,ONLINE'
// COMMAND 'RO *ALL,V 6201,ONLINE'
// COMMAND 'RO *ALL,V 6202,ONLINE'
// COMMAND 'RO *ALL,V 6204,ONLINE'
// COMMAND 'RO *ALL,V 6205,ONLINE'
// COMMAND 'RO *ALL,V 6305,ONLINE'
// COMMAND 'RO *ALL,V 6306,ONLINE'
// COMMAND 'RO *ALL,V 6307,ONLINE'
// COMMAND 'RO *ALL,V 6308,ONLINE'
// COMMAND 'RO *ALL,V 630D,ONLINE'
/** STARTUP PROCEDURE FOR VARYING SAP VOLUMES ONLINE
/**
//ONLINE EXEC PGM=IEFB14

```

Example: A-30 DB2V61H1.VSAMPARM

```
DEFINE CLUSTER -  
  ( NAME(DB2V61H1.DSNDBC.DBH1WRK.DSN4K01.I0001.A001) -  
    LINEAR      -  
    REUSE      -  
    VOLUMES(DB6H01) -  
    RECORDS(6400 50) -  
    SHAREOPTIONS(3 3) ) -  
DATA -  
  ( NAME(DB2V61H1.DSNDBD.DBH1WRK.DSN4K01.I0001.A001) -  
    ) -
```



JCL for Scenario 2 - DFDSS copy with offline DB2

This appendix provides listings of JCL, scripts, and REXX execs that were used in our activities. In some cases, these listings were too extensive to be included in a chapter. In other cases, a script was not part of our procedure, but was a useful adjunct to the work we did.

Example: B-1 RENAMDB2

```
//SAPRES1A JOB (T,0076),SAPRES1,CLASS=A,MSGCLASS=X,  
//          NOTIFY=&SYSUID,REGION=4M  
/*JOBPARM SYSAFF=SC48  
//*          DB2 FLASHCOPY  
//COPYRENA EXEC PGM=ADRDSSU  
//SYSPRINT DD SYSOUT=X  
//SYSUDUMP DD SYSOUT=X,OUTLIM=3000  
//SYSIN    DD *  
    COPY DATASET(INCLUDE(DB2V61K1.DSNDB%.DSNDB06.** -  
                        DB2V61K1.DSNDB%.DSNDB01.** -  
                        DB2V61K1.DSNDB%.DSNRGFDB.** -  
                        DB2V61K1.DSNDB%.DSNRLST.** -  
                        DB2V61K1.DSNDB%.DSNDPSM.**)) -  
    RENAMEU((DB2V61K1.**,DB2V610B.**)) -  
    ALLDATA(*) ALLX TOL(ENQF) -
```

```
NULLMGMTCLAS -  
CATALOG  
/*  
//
```

Example: B-2 SOURED01

```
//SAPRES1A JOB (999,P0K), 'PRINT MAP ', CLASS=A, MSGCLASS=X,  
// NOTIFY=&SYSUID, TIME=1440, REGION=0M  
/*JOBPARM L=9999, SYSAFF=SC48  
//BSDSPRNT EXEC PGM=DSNJU004  
//STEPLIB DD DSN=DSN610.SDSNLOAD, DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DSN=DB2V610B.BSDS01, DISP=SHR  
//SYSIN DD DUMMY  
//
```

Example: B-3 TARBLU01

```
//SAPRES1A JOB (999,P0K), 'BUILD JOBS', CLASS=A, MSGCLASS=X,  
// NOTIFY=&SYSUID, REGION=4M  
/*JOBPARM SYSAFF=SC48  
//*  
//S0104 EXEC PGM=IDCAMS, REGION=4096K  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
DELETE UCAT.VSAPBLU USERCATALOG RECOVERY  
/*
```

Example: B-4 TARBLU02

```
//SAPRES1A JOB (999,P0K), 'INITIALIZE VOLUME', MSGCLASS=X, NOTIFY=&SYSUID  
/*JOBPARM SYSAFF=SC48  
//INIT1 EXEC PGM=ICKDSF, PARM='NOREPLYU'  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
INIT UNIT(3803) VOLID(SAPT21) VFY(SAPT21) -  
PURGE VTOC(1,0,400) SG  
INIT UNIT(3804) VOLID(SAPT22) VFY(SAPT22) -  
PURGE VTOC(1,0,400) SG
```

Example: B-5 TARBLU03

```
//SAPRES1A JOB (999,P0K), 'SAPBLU', CLASS=A, MSGLEVEL=(1,1),
```

```

// MSGCLASS=X,NOTIFY=&SYSUID
/*JOBPARM L=9999,SYSAFF=SC48
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE -
    USERCATALOG -
      (NAME (UCAT.VSAPBLU) -
      IFCATALOG -
      VOLUMES(SAPT27) -
      STORCLAS(SCSAPBLU) -
      STRNO(8) -
      CYLINDERS(100 50 ) -
      SHAREOPTIONS(3 4)) -
    DATA(BUFND(8) -
      NAME (UCAT.VSAPBLU.DATA) -
      CYLINDERS(100 50) -
      CONTROLINTERVALSIZE(4096) -
      FREESPACE(20 20) -
      RECORDSIZE(100 32400)) -
    INDEX(BUFNI(7) -
      NAME (UCAT.VSAPBLU.INDEX) -
      TRACKS(200,50) -
      CONTROLINTERVALSIZE(4096) -
      IMBED -
      REPLICATE)
/*
/** -----
//S0202 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=X
//SYSIN DD *
  DEF ALIAS(NAME(SAPBLU ) RELATE(UCAT.VSAPBLU ) -
  CATALOG(MCAT.OS390R10.V010CAT)
/*

```

Example: B-6 TARBLU04

```

//SAPRES1A JOB (999,P0K), 'PRINT MAP ',CLASS=A,MSGCLASS=X,
// NOTIFY=&SYSUID,TIME=1440,REGION=OM
/*JOBPARM L=9999,SYSAFF=SC48
/**
/*******/
/*******/
/** ***** */
/** DIRECTORY/CATALOG AMS INVOCATION INSTREAM JCL PROCEDURE */
/** ***** */
//DSNTIC EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*

```

```

//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
DELETE DB2V610B.RUNLIB.LOAD
DELETE DB2V610B.SRCLIB.DATA
DELETE DB2V610B.DSNDBC.DSNDB01.DBD01.I0001.A001
DELETE DB2V610B.DSNDBC.DSNDB01.SCT02.I0001.A001

```

Example: B-7 TARBLU05

```

//SAPRES1A JOB (999,POK), 'PRINT MAP ', CLASS=A, MSGCLASS=X,
// NOTIFY=&SYSUID, TIME=1440, REGION=OM
/*JOBPARM L=9999, SYSAFF=SC48
//BSDSPRNO EXEC PGM=DSNJU004
//STEPLIB DD DSN=DSN610.SDSNLOAD, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=DB2V61K1.DBK1.BSDS01, DISP=SHR
//SYSIN DD DUMMY
/*
//BSDSPRN1 EXEC PGM=DSNJU004
//STEPLIB DD DSN=DSN610.SDSNLOAD, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=DB2V61K1.DBK2.BSDS01, DISP=SHR
//SYSIN DD DUMMY
//

```

Example: B-8 TARBLU06

```

//SAPRES1A JOB (999,POK), 'BUILD JOBS', CLASS=A, MSGCLASS=X,
// NOTIFY=&SYSUID, REGION=4M
/*JOBPARM SYSAFF=SC48
/*
/* CHANGE LOG INVENTORY:
/* UPDATE BSDS WITH NEW ACTIVE LOGS
/*
//ACTLOG EXEC PGM=DSNJU003
//STEPLIB DD DISP=SHR, DSN=DSN610.SDSNLOAD
//SYSUT1 DD DISP=OLD, DSN=DB2V610B.BSDS01
//SYSUT2 DD DISP=OLD, DSN=DB2V610B.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
CRESTART CREATE, STARTRBA=B6052793C000, ENDRBA=B6052793C000
/*

```

Example: B-9 TARBLU07

```
-STOP DATABASE(DSNDB06) SPACENAM(SYSTBLSP)
-STOP DATABASE(DSNDB06) SPACENAM(SYSTABLE)
-STOP DATABASE(DBK1WRK)
-STOP DATABASE(DBK2WRK)
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) PARM('RCO') -
LIB('DB2V61K1.RUNLIB.LOAD')
-START DATABASE(DSNDB06) SPACENAM(SYSTBLSP)
-START DATABASE(DSNDB06) SPACENAM(SYSTABLE)
END
//SYSIN DD *
CREATE STOGROUP TEMPSAP VOLUMES("*) VCAT DB2V610B ;
COMMIT ;
ALTER INDEX SAPR3."SYSTABLE~0" USING STOGROUP TEMPSAP;
ALTER INDEX SAPR3."SYSTBLSP~0" USING STOGROUP TEMPSAP;
/*
/**
//DSNTIST EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DB2B)
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) PARM('RCO') -
LIB('DB2V61K1.RUNLIB.LOAD')
END
//SYSIN DD *
DROP DATABASE DBK1WRK ;
DROP DATABASE DBK2WRK ;
```

Example: B-10 TARBLU08

```
//SAPRES1A JOB (999,POK), 'BUILD JOBS', CLASS=A, MSGCLASS=X,
// NOTIFY=&SYSUID, REGION=4M
/*JOBPARM SYSAFF=SC48
/**
//*****/
//DSNTICR EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DB2B)
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) PARM('RCO') -
```

```

        LIB('DB2V61K1.RUNLIB.LOAD')
    END
//SYSIN DD *
    CREATE DATABASE DSNDB07 ;
//*
//*
//DSNTTMP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//*****/
//SYSIN DD *
    DEFINE CLUSTER -
        ( NAME(DB2V610B.DSNDBC.DSNDB07.DSN4K01.I0001.A001) -
          LINEAR -
          REUSE -
          VOLUMES(DB6B02) -
          TRK(900) -
          SHAREOPTIONS(3 3) ) -
    DATA -
        ( NAME(DB2V610B.DSNDBD.DSNDB07.DSN4K01.I0001.A001) -
          ) -

//DSNTIST EXEC PGM=IKJEFT01,DYNAMNBR=20,
// COND=(4,LT,DSNTTMP)
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(DB2B)
    -STOP DATABASE(DSNDB07)
    RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
        LIB('DB2V61K1.RUNLIB.LOAD')
    -START DATABASE(DSNDB07)
    END
//SYSIN DD *
    CREATE TABLESPACE DSN4K01 IN DSNDB07
        BUFFERPOOL BPO
        CLOSE NO
        USING VCAT DB2V610B;
//*

```

Example: B-11 TARBLU09

```

//SAPRES1A JOB (999,P0K), 'BUILD JOBS', CLASS=A, MSGCLASS=X,
// NOTIFY=&SYSUID, REGION=4M
/*JOBPARM SYSAFF=SC48
//*

```

```

/*****/
/*
//DSNTIST EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(DB2B)
    RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) PARM('RC0') -
        LIB('DB2V61K1.RUNLIB.LOAD')
    END
//SYSIN DD *
    CREATE INDEX SAPRES4.INDSTOR
        ON SYSIBM.SYSINDEXPART
        (STORNAME)
        USING STOGROUP TEMPSAP
    ;
    CREATE INDEX SAPRES4.TABSTOR
        ON SYSIBM.SYSTABLEPART
        (STORNAME)
        USING STOGROUP TEMPSAP
    ;
/*-----*/
/* REXX TO EXTRACT STOGROUP NAMES FROM THE TARGET SYSTEM AFTER **/
/* DB2 HAS BEEN RESTARTED. **/
/* PARAMETERS: **/
/* ----- **/
/* 1) REXX PROC TO EXECUTE **/
/* 2) TURN TRACING ON / OFF **/
/* 3) DB2 SUB-SYSTEM TO CONNECT TO **/
/* 4) VCATNAME OF THE NEW HLQ **/
/* 5) CREATOR ID OF EXISTING STOGROUP **/
/* 6) CREATOR ID FOR NEW STOGROUP **/
/* -----*/
//STEP01 EXEC PGM=IKJEFT01,DYNAMNBR=60,REGION=OK,COND=(4,LT),
// PARM='%DB2REXX1 N DB2B DB2V610B RCONWAY SAPRES1'
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
// DD DSN=DB2V61K1.RUNLIB.LOAD,DISP=SHR
//SYSPROC DD DSN=SAPRES5.JCL.SCENAR2.LIBRARY,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DB21OUT1 DD DSN=SAPRES4.TEMP.SDROPSTO,DISP=(,CATLG,DELETE),
// SPACE=(TRK,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB21OUT2 DD DSN=SAPRES4.TEMP.SCREASTO,DISP=(,CATLG,DELETE),
// SPACE=(TRK,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//SYSIN DD DUMMY
//SYSTSIN DD DUMMY

```

```

/**-----**/
/** REXX TO EXTRACT TABLESPACE / INDEXSPACE TO ALTER PRIOR TO **/
/** CHANING THE VCAT NAME. **/
/** PARAMETERS: **/
/** ----- **/
/** 1) REXX PROC TO EXECUTE **/
/** 2) TURN TRACING ON / OFF **/
/** 3) DB2 SUB-SYSTEM TO CONNECT TO **/
/** 4) VCATNAME OF THE NEW HLQ **/
/** 5) CREATOR ID FOR NEW STOGROUP **/
/** 6) CHECK FOR SAP INDICES ON DSNDB06 **/
/** ----- **/
//STEP02 EXEC PGM=IKJEFT01,DYNAMNBR=60,REGION=OK,COND=(4,LT),
// PARM='%DB2REXX2 N DB2B DB2V610B SAPRES1 Y'
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
// DD DSN=DB2V61K1.RUNLIB.LOAD,DISP=SHR
//SYSPROC DD DSN=SAPRES5.JCL.SCENAR2.LIBRARY,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DB21INP1 DD DSN=SAPRES4.TEMP.SDROPSTO,DISP=SHR
/**
//DB22OUT1 DD DSN=SAPRES4.TEMP.SALTPERM,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT2 DD DSN=SAPRES4.TEMP.SALTTEMP,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT3 DD DSN=SAPRES4.TEMP.SSTOPDB,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT4 DD DSN=SAPRES4.TEMP.SSTRADB,DISP=(,CATLG,DELETE),
// SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//SYSIN DD DUMMY
//SYSTSIN DD DUMMY
/**
/**-----**/
/** REXX TO EXTRACT STOGROUP NAMES FROM THE TARGET SYSTEM AFTER **/
/** DB2 HAS BEEN RESTARTED. **/
/** PARAMETERS: **/
/** ----- **/
/** 1) REXX PROC TO EXECUTE **/
/** 2) TURN TRACING ON / OFF **/
/** 3) DB2 SUB-SYSTEM TO CONNECT TO **/
/** 4) VCATNAME OF THE NEW HLQ **/
/** 5) CREATOR ID OF EXISTING STOGROUP **/
/** 6) CREATOR ID FOR NEW STOGROUP **/
/** ----- **/
//STEP03 EXEC PGM=IKJEFT01,DYNAMNBR=60,REGION=OK,
// PARM='%DB2REXX1 N DB2B SAPBLU SAPR3 SAPR3'
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR

```



```

//          DD DSN=DB2V61K1.RUNLIB.LOAD,DISP=SHR
//SYSPROC DD DSN=SAPRES5.JCL.SCENAR2.LIBRARY,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DB21OUT1 DD DSN=SAPRES4.TEMP.DROPSTO,DISP=(,CATLG,DELETE),
//  SPACE=(TRK,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB21OUT2 DD DSN=SAPRES4.TEMP.CREASTO,DISP=(,CATLG,DELETE),
//  SPACE=(TRK,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//SYSIN    DD DUMMY
//SYSTSIN  DD DUMMY
//*
/*-----*//
/* REXX TO EXTRACT TABLESPACE / INDEXSPACE TO ALTER PRIOR TO    *//
/* CHANING THE VCAT NAME.                                         *//
/*  PARAMETERS:                                                  *//
/*  -----                                                       *//
/*  1) REXX PROC TO EXECUTE                                       *//
/*  2) TURN TRACING ON / OFF                                       *//
/*  3) DB2 SUB-SYSTEM TO CONNECT TO                               *//
/*  4) VCATNAME OF THE NEW HLQ                                     *//
/*  5) CREATOR ID FOR NEW STOGROUP                                 *//
/*  6) CHECK FOR SAP INDICES ON DSNDB06                           *//
/*  -----                                                       *//
/*-----*//
//STEP04 EXEC PGM=IKJEFT01,DYNAMNBR=60,REGION=OK,COND=(4,LT),
//  PARM='%DB2REXX2 N DB2B SAPBLU SAPR3 N'
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//          DD DSN=DB2V61K1.RUNLIB.LOAD,DISP=SHR
//SYSPROC DD DSN=SAPRES5.JCL.SCENAR2.LIBRARY,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DB21INP1 DD DSN=SAPRES4.TEMP.DROPSTO,DISP=SHR
//*
//DB22OUT1 DD DSN=SAPRES4.TEMP.ALTPERM,DISP=(,CATLG,DELETE),
//  SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT2 DD DSN=SAPRES4.TEMP.ALTEMP,DISP=(,CATLG,DELETE),
//  SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT3 DD DSN=SAPRES4.TEMP.STOPDB,DISP=(,CATLG,DELETE),
//  SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//DB22OUT4 DD DSN=SAPRES4.TEMP.STRTDB,DISP=(,CATLG,DELETE),
//  SPACE=(CYL,(1,1)),UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//SYSIN    DD DUMMY
//SYSTSIN  DD DUMMY
//*
//
//DSNTIST EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*

```

```

//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(DB2B)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) PARM('RC0') -
    LIB('DB2V61K1.RUNLIB.LOAD')
  END
//SYSIN DD *
  DROP INDEX SAPRES4.INDSTOR ;
  DROP INDEX SAPRES4.TABSTOR ;
/*

```

Example: B-12 TARBLU10

```

//SAPRES1A JOB (999,POK), 'BUILD JOBS', CLASS=A, MSGCLASS=X,
//          NOTIFY=&SYSUID, REGION=4M
/*JOBPARM SYSAFF=SC48
//*
//*****/
//DSNTIST EXEC PGM=IKJEFT01, DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD, DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(DB2B)
  -STOP DATABASE(DSNDB06) SPACENAM(SYSTBLSP)
  -STOP DATABASE(DSNDB06) SPACENAM(SYSTABLE)
//          DD DSN=SAPRES4.TEMP.SSTOPDB, DISP=SHR
//          DD *
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
    LIB('DB2V61K1.RUNLIB.LOAD')
  -START DATABASE(DSNDB06) SPACENAM(SYSTBLSP)
  -START DATABASE(DSNDB06) SPACENAM(SYSTABLE)
  END
//*
//SYSIN DD DSN=SAPRES4.TEMP.SALTTEMP, DISP=SHR
//          DD DSN=SAPRES4.TEMP.SDROPSTO, DISP=SHR
//          DD DSN=SAPRES4.TEMP.SCREASTO, DISP=SHR
//          DD DSN=SAPRES4.TEMP.SALTPERM, DISP=SHR
//          DD *
  ALTER INDEX SAPR3."SYSTABLE~0" USING STOGROUP SYSDEFLT ;
  ALTER INDEX SAPR3."SYSTBLSP~0" USING STOGROUP SYSDEFLT ;
  COMMIT;
  DROP STOGROUP TEMPSAP ;
//*
//DSNTIST EXEC PGM=IKJEFT01, DYNAMNBR=20, COND=(9,LT)
//STEPLIB DD DSN=DSN610.SDSNLOAD, DISP=SHR
//SYSTSPRT DD SYSOUT=*

```

```

//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(DB2B)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA61) -
    LIB('DB2V61K1.RUNLIB.LOAD')
//      DD DSN=SAPRES4.TEMP.STRTDB,DISP=SHR
//      DD *
  END
//*
//SYSIN DD DSN=SAPRES4.TEMP.ALTTEMP,DISP=SHR
//      DD DSN=SAPRES4.TEMP.DROPSTO,DISP=SHR
//      DD DSN=SAPRES4.TEMP.CREASTO,DISP=SHR
//      DD DSN=SAPRES4.TEMP.ALTPERM,DISP=SHR
//*

```

Example: B-13 TARBLU11

```

//SAPRES1A JOB (999,POK), 'BUILD JOBS',CLASS=A,MSGCLASS=X,
//      NOTIFY=&SYSUID,REGION=4M
/*JOBPARM SYSAFF=SC48
//*
//DSNTIST EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=DSN610.SDSNLOAD,DISP=SHR
//SYSTSPT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
  DSN SYSTEM(DB2B)
  RUN PROGRAM(DSNTPE2) PLAN(DSNTPE61) -
    LIB('DB2V61K1.RUNLIB.LOAD')
  END
//*
//SYSIN DD *
SELECT NAME,VCATNAME
       FROM SYSIBM.SYSSTOGROUP
       WHERE VCATNAME = 'DB2V61K1'
          OR VCATNAME = 'SAPRED' ;
SELECT TSNAME,STORNAME,VCATNAME,DBNAME
       FROM SYSIBM.SYSTABLEPART
       WHERE VCATNAME = 'DB2V61K1'
          OR VCATNAME = 'SAPRED' ;
SELECT IXNAME,STORNAME,VCATNAME
       FROM SYSIBM.SYSINDEXPART
       WHERE VCATNAME = 'DB2V61K1'
          OR VCATNAME = 'SAPRED' ;

```



C

REXX procedures

This appendix lists the REXX procedures that were written to list the DB2 objects requiring modification after the completion of the copy of the SAP DB2 system from the source to the target system. The first procedure extracts the STOGROUPs to be ALTERed and the second extracts all the TABLESPACEs and INDEXes to be modified.

These procedures use the DB2 REXX interface which is activated by installing FMID JDB661H for DB2 V6.1. Included in the install are binds of the plans/packages required to interface to DB2.

The JCL to execute the following procedures is found in Example A-23 on page 83.

Example: C-1 REXX procedure DB2REXX1

```
/* Language      : REXX                                     */
/* -----*/
/* This REXX program interrogates the DB2 CAT/DIR to extract all
/* SAP STOGROUPS names.                                     */
/*                                                         */
/* Output is a series of files containing DDL to DROP and re-CREATE
/* STOGROUPS with a new HLQ.                               */
/*                                                         */
/* Input is a) debug option                               */
```

```

/*          b) DB2 system to connect to          */
/*          c) new vcat HLQ                       */
/*          d) stogroup creator id (exisiting creator id) */
/*          e) creator id for new stogroup (new creator id) */
/*          */
/* -----*/

```

```

TRACE n
PARSE ARG debug db2ssid newcat stoid creatid

```

```

IF debug = 'Y' then TRACE i

```

```

/*****/
/** Establish Link to DB2. First check if exists, else setup.      **/
/*****/

```

```

'SUBCOM DSNREXX'

```

```

IF RC

```

```

  THEN DO

```

```

    SRC=RXSUBCOM('ADD','DSNREXX','DSNREXX')

```

```

    IF SRC <> 0

```

```

      THEN DO

```

```

        SAY '+*****+'

```

```

        SAY '+*   LINK TO DB2 INTERFACE FAILED           *+'

```

```

        SAY '+*   RETURN CODE IS ' SRC '               *+'

```

```

        SAY '+*****+'

```

```

      EXIT RC = SRC

```

```

    END

```

```

  END

```

```

/*****/
/** Connect to the required DB2 Subsystem.                      **/
/*****/

```

```

ADDRESS DSNREXX

```

```

"CONNECT " db2ssid

```

```

IF RC <> 0

```

```

  THEN DO

```

```

    SAY '+*****+'

```

```

    SAY '+*   CONNECTION TO DB2 FAILED                 *+'

```

```

    SAY '+*   RETURN CODE IS ' RC '                   *+'

```

```

    SAY '+*****+'

```

```

  EXIT RC = RC

```

```

  END

```

```

/*****/
/** Setup statements for the CURSOR                               **/
/*****/

```

```

Cursql = "  SELECT NAME, VCATNAME " || " ,
          " FROM SYSIBM.SYSSTOGROUP " || " ,
          " WHERE CREATOR = " || "||stoid||" " || " ,
          " ORDER BY NAME "

```

```

/*****
/** Commands to Declare, Prepare, Open the Cursor          **/
*****/
"EXECSQL DECLARE C1 CURSOR FOR S1"
IF SQLCODE <> 0
  THEN
    CALL SQLERROR

"EXECSQL PREPARE S1 FROM :Cursq1"
IF SQLCODE <> 0
  THEN
    CALL SQLERROR

"EXECSQL OPEN C1"
IF SQLCODE <> 0
  THEN
    CALL SQLERROR

/*****
/** Main routine to retrieve the required DB2 data          **/
*****/
$count1 = 1
DO SQLCODE = 0 TO 100
  "EXECSQL FETCH C1 INTO :stoname, :catname"
  IF SQLCODE = 100 THEN RC = 0
  IF SQLCODE <> 0 & SQLCODE <> 100
    THEN
      CALL SQLERROR
  IF SQLCODE = 100
    THEN LEAVE

  IF SQLCODE <> 100
    THEN DO

    /* output fields in correct columns */

    $out1.$count1 = ' DROP STOGROUP ' || stoname || ';'
    $out2.$count1 = ' CREATE STOGROUP ' || stoname ||
      ' VOLUMES("*) VCAT ' || newcat || ' ' ;'
    $count1 = $count1 + 1
  END
END
/*****
/** Disconnect from DB2 Subsystem                          **/
*****/
S_RC=RXSUBCOM('DELETE', 'DSNREXX', 'DSNREXX')
IF RC <> 0
  THEN CALL Error

```

```

/*****
/** Write STOGROUP DROP statements to a file          **/
/*****
ADDRESS TSO
DO $count2 = 1 to $count1 - 1
    NEWSTACK
    PUSH $out1.$count2
    'EXECIO 1 DISKW db21out1'
    DELSTACK
END
'EXECIO 0 DISKW db21out (FINIS'
/*****
/** Write STOGROUP CREATE statements to a file          **/
/*****
/* Set SAP userid */
$out5.1 = " SET CURRENT SQLID=" || creatid || " ";
$out5.2 = ' COMMIT ;'

DO $count5 = 1 to 2
    NEWSTACK
    PUSH $out5.$count5
    'EXECIO 1 DISKW db21out2'
    DELSTACK
END
/*
DO $count2 = 1 to $count1 - 1
    NEWSTACK
    PUSH $out2.$count2
    'EXECIO 1 DISKW db21out2'
    DELSTACK
END
'EXECIO 0 DISKW db22out (FINIS'

EXIT RC = RC
/*****
/** Errors          **/
/*****
SQLERROR:
SAY '+*****+'
SAY '+* DB2 ERROR ACCESSING THE DATABASE *+'
SAY '+* ERROR CODE IS ' SQLCODE ' !!! *+'
SAY '+*****+'
SAY SQLERRMC
SAY SQLERRP
SAY 'ERROR CODE ...' SQLERRD.1 ' , 'SQLERRD.2' , 'SQLERRD.3
SAY ' ' SQLERRD.4 ' , 'SQLERRD.5' , 'SQLERRD.6
SAY ' ' SQLERRD.0
SAY 'WARNING CODE..' SQLWARN.1 ' , 'SQLWARN.2' , 'SQLWARN.3

```



```

SAY '          ' SQLWARN.4 ' , 'SQLWARN.5' , 'SQLWARN.6
EXIT RC = RC
/*****
/** On Error Processing          **/
/*****
Error:
  SAY '>>> Error on line =' SIGL ' RC = ' RC ' <<<'
EXIT RC

```

Example: C-2 REXX procedure DB2REXX2

```

/* Language      : REXX                      */
/* -----*/
/* This REXX program reads the input from DB2REXX1 to extract all
/* SAP tablespaces and indices linked to STOGROUPS.
/*
/*
/* Output is a series DDL statements to ALTER tablespaces and
/* indices to temp STOGROUPS. A second output will ALTER the
/* tablespaces and indices back tp the original STOGROUPS.
/*
/*
/* Input is a) debug option
/*          b) DB2 system to connect to
/*          c) vcatname for ALTERs
/*          d) creator id for new stogroups
/*          e) check for SAP indices on DSNDB06
/*
/*
/* -----*/

```

```

TRACE n
PARSE ARG debug db2ssid newcat creatid sapix

```

```

IF debug = 'Y' then TRACE i

```

```

'EXECIO * DISKR db21inp1 (STEM $input1. FINIS)'
$count4 = $input1.0
/*****
/** Establish Link to DB2. First check if exists, else setup.          **/
/*****
'SUBCOM DSNREXX'
IF RC
  THEN DO
    SRC=RXSUBCOM('ADD','DSNREXX','DSNREXX')
    IF SRC <> 0
      THEN DO
        SAY '+*****+'
        SAY '+*   LINK TO DB2 INTERFACE FAILED          *+'
        SAY '+*   RETURN CODE IS ' SRC '              *+'
        SAY '+*****+'

```

```

        EXIT RC = SRC
    END
END
/*****
/** Connect to the required DB2 Subsystem.          **/
*****/
ADDRESS DSNREXX
"CONNECT " db2ssid
IF RC <> 0
    THEN DO
        SAY '+*****+'
        SAY '+*   CONNECTION TO DB2 FAILED           *+'
        SAY '+*   RETURN CODE IS ' RC '           *+'
        SAY '+*****+'
        EXIT RC = RC
    END
/*****
/** Main routine to retrieve the required DB2 data    **/
*****/
$count2 = 1
$count3 = 1
$count6 = 1
$count7 = 1
$count8 = 1

DO $count1 = 1 to $count4
                                                    /* Extract STOGROUP name */
    stoname1 = SUBSTR($input1.$count1,16,8)

    /*****
    /** Setup statements for the CURSOR                **/
    *****/
    Cursq11 = "  SELECT STRIP(DBNAME), TSNAME, PARTITION " || ,
              "  FROM SYSIBM.SYSTABLEPART " || ,
              "  WHERE STORNAME = " || stoname1 || " " || ,
              "  ORDER BY TSNAME, PARTITION DESC "

    Cursq12 = "  SELECT STRIP(IXCREATOR), IXNAME, PARTITION " || ,
              "  FROM SYSIBM.SYSINDEXPART " || ,
              "  WHERE STORNAME = " || stoname1 || " " || ,
              "  ORDER BY IXNAME, PARTITION DESC "

    Cursq13 = "  SELECT STRIP(IXCREATOR), IXNAME, STORNAME " || ,
              "  FROM SYSIBM.SYSINDEXPART A, " || ,
              "  SYSIBM.SYSINDEXES B " || ,
              "  WHERE B.DBNAME = 'DSNDB06' " || ,
              "  AND A.IXNAME = B.NAME " || ,
              "  AND A.IXCREATOR ^= 'SYSIBM' " || ,
              "  ORDER BY A.IXNAME "

```

```

/*****
/** Commands to Declare, Prepare, Open the Cursor      **/
*****/
"EXECSQL DECLARE C1 CURSOR FOR S1"
IF  SQLCODE <> 0
  THEN
    CALL SQLERROR

"EXECSQL PREPARE S1 FROM :Cursq11"
IF  SQLCODE <> 0
  THEN
    CALL SQLERROR

"EXECSQL OPEN C1"
IF  SQLCODE <> 0
  THEN
    CALL SQLERROR

"EXECSQL DECLARE C2 CURSOR FOR S2"
IF  SQLCODE <> 0
  THEN
    CALL SQLERROR

"EXECSQL PREPARE S2 FROM :Cursq12"
IF  SQLCODE <> 0
  THEN
    CALL SQLERROR

"EXECSQL OPEN C2"
IF  SQLCODE <> 0
  THEN
    CALL SQLERROR
/*****

DO SQLCODE = 0 TO 100
  "EXECSQL FETCH C1 INTO :dbname1, :tsname1, :tspart  "
  IF  SQLCODE = 100 THEN RC = 0
  IF  SQLCODE <> 0 & SQLCODE <> 100
    THEN
      CALL SQLERROR
  IF  SQLCODE = 100
    THEN LEAVE

IF  SQLCODE <> 100
  THEN DO

  /* output fields in correct columns */
  /* output with original STOGROUPS */

```

```

IF tspart > 0
THEN DO
$out1.$count2 = ' ALTER TABLESPACE ' ||
                dbname1 || '.' ||
                tsname1 || ' PART ' || tspart ||
                ' USING STOGROUP ' || stoname1 || ' ;'
$out2.$count6 = ' ALTER TABLESPACE ' ||
                dbname1 || '.' ||
                tsname1 || ' PART ' || tspart ||
                ' USING STOGROUP TEMPSAP ;'
$count6 = $count6 + 1
$count2 = $count2 + 1
$count7 = $count7 + 1
END
ELSE DO
$out1.$count2 = ' ALTER TABLESPACE ' ||
                dbname1 || '.' ||
                tsname1 || ' USING STOGROUP ' ||
                stoname1 || ' ;'
$out2.$count6 = ' ALTER TABLESPACE ' ||
                dbname1 || '.' ||
                tsname1 || ' USING STOGROUP TEMPSAP ;'
$count6 = $count6 + 1
$count2 = $count2 + 1
$count7 = $count7 + 1
END

/* commit after 500 ALTER's          */

IF $count7 > 500
THEN DO
    $out1.$count2 = ' COMMIT ;'
    $out2.$count6 = ' COMMIT ;'
    $count2 = $count2 + 1
    $count6 = $count6 + 1
    $count7 = 0
END

END

END

DO SQLCODE = 0 TO 100
"EXECSQL FETCH C2 INTO :ixcreat1, :ixname1, :ixpart "
IF SQLCODE = 100 THEN RC = 0
IF SQLCODE <> 0 & SQLCODE <> 100
THEN
    CALL SQLERROR

```

```

IF SQLCODE = 100
  THEN LEAVE

IF SQLCODE <> 100
  THEN DO

  /* output fields in correct columns */
  /* double quotes required due SAP index naming standards */

IF ixpart > 0
  THEN DO
  $out1.$count2 = ' ALTER INDEX ' ||
                  ixcreat1 || '.' ||
                  ixname1 || ' PART ' || ixpart ||
                  ' USING STOGROUP ' || stoname1 || ';'
  $out2.$count6 = ' ALTER INDEX ' ||
                  ixcreat1 || '.' ||
                  ixname1 || ' PART ' || ixpart ||
                  ' USING STOGROUP TEMPSAP ;'

  $count6 = $count6 + 1
  $count2 = $count2 + 1
  $count7 = $count7 + 1
  END
ELSE DO
  $out1.$count2 = ' ALTER INDEX ' ||
                  ixcreat1 || '.' ||
                  ixname1 || ' USING STOGROUP ' ||
                  stoname1 || ';'
  $out2.$count6 = ' ALTER INDEX ' ||
                  ixcreat1 || '.' ||
                  ixname1 || ' USING STOGROUP TEMPSAP ;'

  $count6 = $count6 + 1
  $count2 = $count2 + 1
  $count7 = $count7 + 1
  END

/* commit after 500 ALTER's */

IF $count7 > 500
  THEN DO
  $out1.$count2 = ' COMMIT ;'
  $out2.$count6 = ' COMMIT ;'
  $count2 = $count2 + 1
  $count6 = $count6 + 1
  $count7 = 0
  END

END

```

```

END

IF sapix = 'Y'
  THEN DO
    "EXECSQL DECLARE C3 CURSOR FOR S3"
    IF SQLCODE <> 0
      THEN
        CALL SQLERROR

    "EXECSQL PREPARE S3 FROM :Cursq13"
    IF SQLCODE <> 0
      THEN
        CALL SQLERROR

    "EXECSQL OPEN C3"
    IF SQLCODE <> 0
      THEN
        CALL SQLERROR

    DO SQLCODE = 0 TO 100
      "EXECSQL FETCH C3 INTO :ixcreat1, :ixname1, :ixsto "
      sapix = 'N'

      IF SQLCODE = 100 THEN RC = 0
      IF SQLCODE <> 0 & SQLCODE <> 100
        THEN
          CALL SQLERROR
      IF SQLCODE = 100
        THEN LEAVE

      IF SQLCODE <> 100
        THEN DO
          $out1.$count2 = ' ALTER INDEX ' || ,
            ixcreat1 || '.' || ,
            ixname1 || ' USING STOGROUP ' || ,
            ixsto || ' ;'
          $count2 = $count2 + 1
          $out2.$count6 = ' ALTER INDEX ' || ,
            ixcreat1 || '.' || ,
            ixname1 || ' USING STOGROUP TEMPSAP ;'
          $count6 = $count6 + 1
        END
      END

    "EXECSQL CLOSE C3"
    IF SQLCODE <> 0
      THEN

```

```

        CALL SQLERROR
    END
/*****
/** Close the open Cursors          **/
*****/
"EXECSQL CLOSE C1"
IF SQLCODE <> 0
    THEN
        CALL SQLERROR

"EXECSQL CLOSE C2"
IF SQLCODE <> 0
    THEN
        CALL SQLERROR

END
/*****
/** Disconnect from DB2 Subsystem  **/
*****/
S_RC=RXSUBCOM('DELETE','DSNREXX','DSNREXX')
IF RC <> 0
    THEN CALL Error
/*****
/** Write temp ALTER statements     **/
*****/
ADDRESS TSO
                                /* Create temp STOGROUP */
$out5.1 = " SET CURRENT SQLID=" || creatid || " ";
$out5.2 = ' CREATE STOGROUP TEMPSAP VOLUMES(" ") ' ||          ,
        ' VCAT ' || newcat || ' ';
$out5.3 = ' COMMIT ;'

DO $count5 = 1 to 3
    NEWSTACK
    PUSH $out5.$count5
    'EXECIO 1 DISKW db22out2'
    DELSTACK

END
/*
DO $count5 = 1 to $count2 - 1
    NEWSTACK
    PUSH $out2.$count5
    'EXECIO 1 DISKW db22out2'
    DELSTACK

END
/*
'EXECIO 0 DISKW db22out1 (FINIS'
/*****
/** Write original ALTER statements to a file          **/
*****/

```

```

/*****/
/* Set SAP userid */
$out5.1 = " SET CURRENT SQLID=' ' || creatid || ' ' ;"
$out5.2 = ' COMMIT ;'

DO $count5 = 1 to 2
  NEWSTACK
  PUSH $out5.$count5
  'EXECIO 1 DISKW db22out1'
  DELSTACK
END
/* */
DO $count5 = 1 to $count6 - 1
  NEWSTACK
  PUSH $out1.$count5
  'EXECIO 1 DISKW db22out1'
  DELSTACK
END
/* */
$out5.1 = ' COMMIT ;'
$out5.2 = ' DROP STOGROUP TEMPSAP ; '
DO $count5 = 1 to 2
  NEWSTACK
  PUSH $out5.$count5
  'EXECIO 1 DISKW db22out1'
  DELSTACK
END
/* */
'EXECIO 0 DISKW db22out2 (FINIS'

/*****/
/** Write STOP DATABASE commands **/
/*****/
$out3 = ' -STOP DATABASE(*) SPACENAM(*) '
NEWSTACK
PUSH $out3
'EXECIO 1 DISKW db22out3'
DELSTACK
'EXECIO 0 DISKW db22out3 (FINIS'

/*****/
/** Write START DATABASE commands **/
/*****/
$out4 = ' -START DATABASE(*) SPACENAM(*) '
NEWSTACK
PUSH $out4
'EXECIO 1 DISKW db22out4'
DELSTACK
'EXECIO 0 DISKW db22out4 (FINIS'

```



```

EXIT RC = RC
/*****
/** Errors **
/*****
SQLERROR:
  SAY '+*****+'
  SAY '+*   DB2 ERROR ACCESSING THE DATABASE   *+'
  SAY '+*   ERROR CODE IS ' SQLCODE '      !!!   *+'
  SAY '+*****+'
  SAY SQLERRMC
  SAY SQLERRP
  SAY 'ERROR CODE ...' SQLERRD.1 ' , 'SQLERRD.2' , 'SQLERRD.3
  SAY '          ' SQLERRD.4 ' , 'SQLERRD.5' , 'SQLERRD.6
  SAY '          ' SQLERRD.0
  SAY 'WARNING CODE..' SQLWARN.1 ' , 'SQLWARN.2' , 'SQLWARN.3
  SAY '          ' SQLWARN.4 ' , 'SQLWARN.5' , 'SQLWARN.6
EXIT RC = RC
/*****
/** On Error Processing **
/*****
Error:
  SAY '>>> Error on line =' SIGL ' RC = ' RC ' <<<'
EXIT RC

```



D

Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

<ftp://www.redbooks.ibm.com/redbooks/SG246287>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the redbook form number, SG246287.

Using the Web material

The additional Web material that accompanies this redbook includes the following zipped file containing sample JCL and REXX execs:

<i>File name</i>	<i>Description</i>
hsc.zip	Sample JCL code and REXX execs for Scenario 1 and 2

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder. The contents can be viewed using NOTEPAD:

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 121.

- ▶ *SAP R/3 on DB2 for OS/390: Implementing with AIX or Windows NT Application Servers*, SG24-4945
- ▶ *SAP R/3 on DB2 UDB for OS/390: OS/390 Application Server*, SG24-5840
- ▶ *IBM Enterprise Storage Server*, SG24-5465

Other resources

These publications are also relevant as further information sources:

- ▶ *DB2 UDB for OS/390 Data Sharing: Planning and Administration Version 6* SC26-9007
- ▶ *DB2 for OS/390 Administration Guide*, SC26-9003
- ▶ *DB2 for OS/390 Utility Guide and Reference*, SC26-9015
- ▶ *R/3 Installation on OS/390 UNIX System Services: DB2 UDB for OS/390*, Material Number 51013703
- ▶ *R/3 Homogeneous System Copy*, Material Number 51009474

Referenced Web sites

This Web site is also relevant as further information source:

<http://www.storage.ibm.com/hardsoft/diskdr1s/technology.htm>

How to get IBM Redbooks

Search for Redbooks or Redpieces, view, download, or order hardcopy from the net at:

<http://www.ibm.com/redbooks>

Also download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Special notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Abbreviations and acronyms

ACS	access control system	ESCON	enterprise systems connection (architecture, IBM S/390)
AIX	Advanced Interactive Executive (an IBM version of the UNIX operating system)	ESS	Enterprise Storage System (IBM System/390)
APAR	authorized program analysis report	FDDI	Fiber Distributed Data Interface (100 Mbits/s fiber optic LAN)
BSDS	bootstrap dataset (DB2)	FICON	fiber connection
CF	Coupling Facility	Gb	gigabit (one Gb = 10 ⁹ bits)
CPU	central processing unit	GB	gigabyte (one GB = 10 ⁹ bytes)
DASD	direct access storage device	GBP	group buffer pool
DB2	Database 2 (an IBM relational database management system)	HA	high availability
DB2 UDB for OS/390	Database 2 universal database for OS/390 (OS/390 implementation of the DB2 product)	HACMP	high availability cluster multi-processing (AIX)
DBA	database administrator	HFS	hierarchical file systems
DBD	database descriptor	I/O	input output
DBMS	database management system	IBM	International Business Machines Corporation
DDIC	data dictionary	ICF	1. integrated catalog facility (OS/390) 2. internal coupling facility (S/390 architecture)
DDL	data definition language	ICLI	integrated call level interface
DML	data manipulation language	ID	identifier
DR	disaster recovery	IDCAMS	the program name for access method services (OP SYS)
EENQ	enqueue	IPL	initial program load
ERP	enterprise resource planning		

IRLM	integrated resource lock manager	SAP AG	SAP Aktiengesell (German name of software vendor)
IT	information technology	SAP R/3	SAP Release 3 (software product of SAP)
ITSO	International Technical Support Organization	SD	Sales and Distribution (SAP application)
JCL	job control language	SMP/E	system modification program/extended (MVS)
LAN	local area network	SMS	System Managed Storage
LPAR	logical partition	SPOF	single point of failure
LRSN	log record sequence number	SQL	structured query language
LUW	logical unit of work	TB	terabyte (one TB = 10 ¹² bytes)
MVS	multiple virtual storage (IBM System 370 & 390)	TCP/IP	Transmission Control Protocol/Internet Protocol
OEM	original equipment manufacture	UDB	universal database
OS/390	Operating System (for the) IBM System/390	UNIX	An operating system developed at Bell Laboratories
OSA	open systems adapter (IBM System/390)	USS	UNIX System Services
OSS	online support service (SAP)	VSAM	Virtual Storage Access Method (IBM)
PPRC	peer-to-peer remote copy	WLM	Workload Manager
PTF	program temporary fix	XES	cross-system extended services (MVS)
PU	processor unit		
R/W	read-write		
RACF	Resource Access Control Facility		
RBA	relative byte address		
RFC	remote function call		
RID	Record IDentifier (DB2)		
RVA	RAMAC Virtual Array		
SAP	1.Systems, Applications, Products in Data Processing (software vendor) 2.system assist processor (S/390 architecture)		

Index

A

ABAP 125
AIX 125
application testing and quality assurance 3

B

backup 30, 53, 63
BSDS 16
 skeleton 16

C

checking the source environment 27, 35, 58
cold start 6, 8, 55, 59
conditional restart 6, 7, 47
copying the source system 28, 36, 59

D

data mining 5
DB2 system cloning 15
 designing 23
 phases 27
 checking the source environment 27
 copying the source system 28
 preparing the target disk environment 29
 restarting target DB2 system 29
 restarting target SAP system 30
 restoring and building target disks 29
 verifications and backup 30
 planning 15
designing DB2 system cloning 23
 configuration considerations 24
 data sharing to data sharing 24
 data sharing to non-data sharing 26
 non-data sharing to data sharing 27
 non-data sharing to non-data sharing 27
DFDSS copy with offline DB2 55
dumpconditioning 35

E

EMC Timefinder 7
ESS Flashcopy 7

F

Flashcopy 34, 35
Flashcopy with suspended DB2 33

H

homogeneous system copy 2
 DFDSS copy 56
 Flashcopy 34
 implementation steps 13
 methodologies 6
 choosing a method 8
 offline copy methodology 7
 online copy methodology 6
Scenario 1 33
Scenario 2 55

I

implementation steps 13
installation customizations 51

J

JCL 65, 93

L

log suspend 6, 37
 PTF 16

M

methodologies 6

O

offline copy 7
online copy 6

P

planning for DB2 system cloning 15
 offline copy 20
 online copy 16
preparing the target disk environment 29, 38, 57
production maintenance 4

R

- Redbooks Web site
 - Contact us x
- rename system 11
- reporting 5
- restarting target DB2 system 29, 42, 44, 59
- restarting target SAP system 30, 62
- restoring and building target disks 29, 41
- REXX 105
 - FMID 105
- RVA Snapshot 7

S

- Scenario 1- Flashcopy with suspended DB2 10, 33
- Scenario 2 - DFDSS copy with offline DB2 12, 55
- source system 10
- storage group 16
 - SMS 16
 - UCAT 16
- system copy 2
 - heterogeneous 2
 - homogeneous 2
 - why HSC? 3
- system function testing 4

T

- target system 11, 12
- test scenarios 10
 - DFDSS copy with offline DB2 12
 - Flashcopy with suspended DB2 10
- timing 16
- training 5

V

- verifications and backup 30, 53, 63

W

- why HSC?
 - application testing and quality assurance 3
 - data mining 5
 - production maintenance 4
 - reporting 5
 - system function testing 4
 - training 5



SAP on DB2 for z/OS and OS/390: DB2 System Cloning

(0.2" spine)
0.17" <-> 0.473"
90 <-> 249 pages



SAP on DB2 for z/OS and OS/390: DB2 System Cloning



Redbooks

Hands-on scenarios to create test databases

Online FlashCopy method for ESS disks

Offline DFDSS copy/rename method

This IBM Redbook shows the different techniques customers can use to clone their DB2 system. Although the scenarios in the book use an SAP workload, the DB2 cloning techniques we describe are not specific to SAP. These techniques can apply to any DB2 system and the applications running on that DB2.

The book describes hands-on details of selected DB2 database cloning scenarios that are typically the most popular techniques currently used by customers, as well as the most efficient scenarios we recommend to DB2 database administrators.

These scenarios are based on SAP examples. Our source system resides on a DB2 data sharing group, and we have two target systems: one resides on another DB2 data sharing group, and one resides on a non-data sharing DB2. All source and target systems are in the same OS/390 Sysplex environment.

We discuss and show DB2 system cloning techniques using the following methods:

- ESS FlashCopy
- DFSMS copy and rename

The book addresses an audience with in-depth knowledge of DB2 for z/OS and OS/390 database administration and familiarity with SAP system requirements.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks