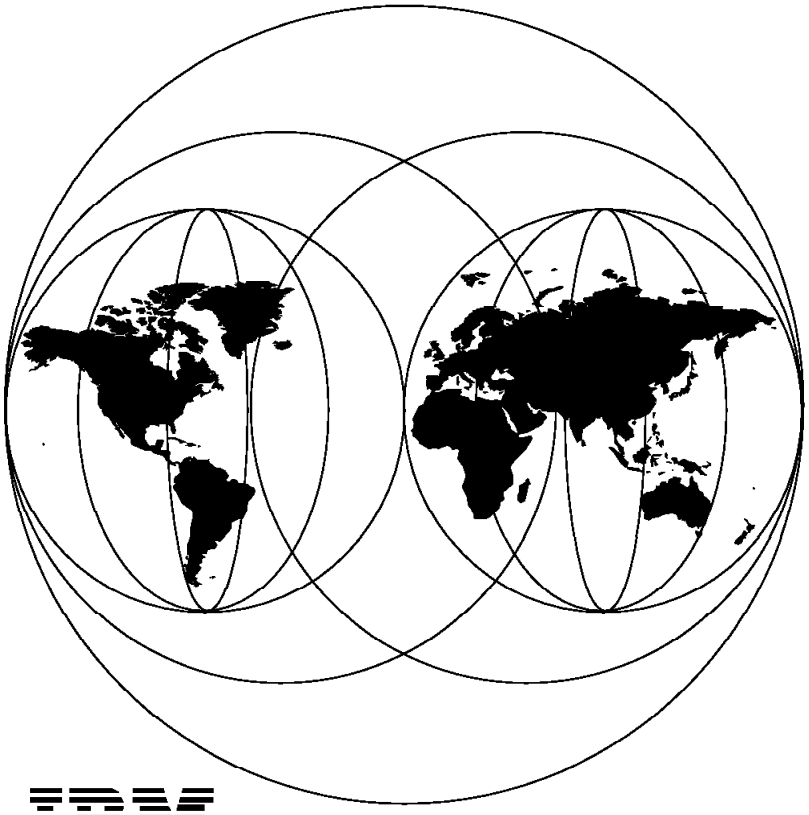
SG24-2039-00

DB2 Recovery on VSE and VM Using the Data Restore Feature

March 1997



IBM
International Technical Support Organization
Boeblingen Center

International Technical Support Organization

SG24-2039-00

DB2 Recovery on VSE and VM Using the Data Restore Feature

March 1997

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, “Special Notices” on page 111.

First Edition (March 1997)

This edition applies to Version 5 Release 1 of Database/2 (DB2) Data Restore Feature, Program Number 5648-158, for use with the VM/ESA or VSE/ESA operating systems and makes reference to Version 3 Release 5 of Structured Query Language/Data System (SQL/DS) Data Restore Feature.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. 3222 Building 71032-02
Postfach 1380
71032 Böblingen, Germany

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.

Contents

Figures

Tables

Preface

Using the possibilities offered by the Data Restore feature, customers can reduce their archiving time, effort, and resources. Online archives can replace offline user archives, increasing the availability of the database. Duplicate backups, which currently are often practiced, can be avoided, while at the same time the recovery capabilities are improved.

This redbook provides detailed information on the usage of the DB2 Data Restore feature. It describes scenarios and gives recommendations related to the backup and restore of a DB2 database, partially or completely. Procedures are documented so that they can easily be executed for test purposes. It points you to all the facts and provides valuable information that you will need to develop, appraise or update your own DB2 recovery strategy.

The document was written for database administrators and for all those people who are responsible for controlling the availability of backups in case something happens to your database and you need to do a full or partial restore. Some knowledge of DB2 database administration is assumed, as well as a basic knowledge of either VM/ESA or VSE/ESA.

The Team that Wrote this Redbook

This redbook was produced and updated during two residency projects each by a team of specialists from around the world working at the International Technical Support Organization Böblingen Center.

Eberhard Lange (LANGE at BOEVM3), from the International Technical Support Organization Böblingen Center, was the project leader.

Zbigniew Przytula (PRZYTULG at BRUVMIS1), from IBM Belgium.

Veronika Wagner (WGN at SDVM1), from the IBM Laboratory in Böblingen.

Marcelo Nicolozzi (NICOLOZI at SPOVMSIB), from IBM Brazil.

Sergio Dill (dill@super.unijui.tche.br), from the Ijui University Brazil.

Christine Jones (CJONES at ISSCAUS), from ISSC Australia.

What is new in this Update

This Redbook is an updated version of *SQL/DS Archive and Recovery Using the Data Restore Feature*, SG24-4833. It describes the new functional enhancements included in the Data Restore Feature of DB2 Server for VSE and VM Version 5 Release 1.

A new section was added elaborating on the storage pool level recovery offered through the improved RESTORE command, and reflects this capability also in the strategy discussion and the time measurement sections.

Additional improvements include more emphasis on the difference between LOGMODEs A and L with regard to checkpoints and the resulting value of backlevel online archives, which also affects the recovery strategy.

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 127 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Home Pages at the following URLs:

  For Internet users http://www.redbooks.ibm.com
  For IBM Intranet users http://w3.itso.ibm.com/redbooks

- Send us a note at the following address:

  redbook@vnet.ibm.com

Chapter 1. Introduction to the Data Restore Feature

This chapter provides an overview of the possibilities and restrictions of the Data Restore feature available with DB2 Server for VSE & VM V5R1. It offers positioning, presents its functions and some remarks regarding the command syntax.

1.1 Positioning

The Data Restore feature, first available with SQL/DS V3R5, supports both the VM/ESA and the VSE/ESA environments. It provides the DB2 customer with new granular options to back up and restore a database machine in case of any system failure, user programming error or user logic error. The Data Restore feature extends the ways to back up or restore the whole database and provides the option to recover parts of it.

Prior to SQL/DS V3R5, customers could use DB2 recovery facilities to recover the whole database or they could use tools provided with the operating systems, such as DDR, DFSMS, Fastcopy or VSAM Backup/Restore. Or, customers could choose other user or vendor applications to do a complete backup and restore of the database.

There are cases where only a part of the database, such as a table, needs to be restored. DB2 Database Services Utility (DBSU) could be used to make backup copies of tables or dbspaces and to restore one or more tables from those backups.

Previously, customers could not restore a table from a DB2 database archive so they often took duplicate backups of critical tables. However, they could not selectively apply the log records to these tables when they were restored. Forward log recovery using the log records was impossible.

Data Restore lets you recover a single or a set of storage pools from a full archive.

The Data Restore feature lets you reload tables from a full archive and avoids duplicate selective backups with Database Services Utility or other utilities.

Data Restore also lets you reload a table and then apply the log to that specific table. Also you can choose whether to apply all the log records or apply only the log records to a certain point in time.

1.2 DB2 Recovery Concepts

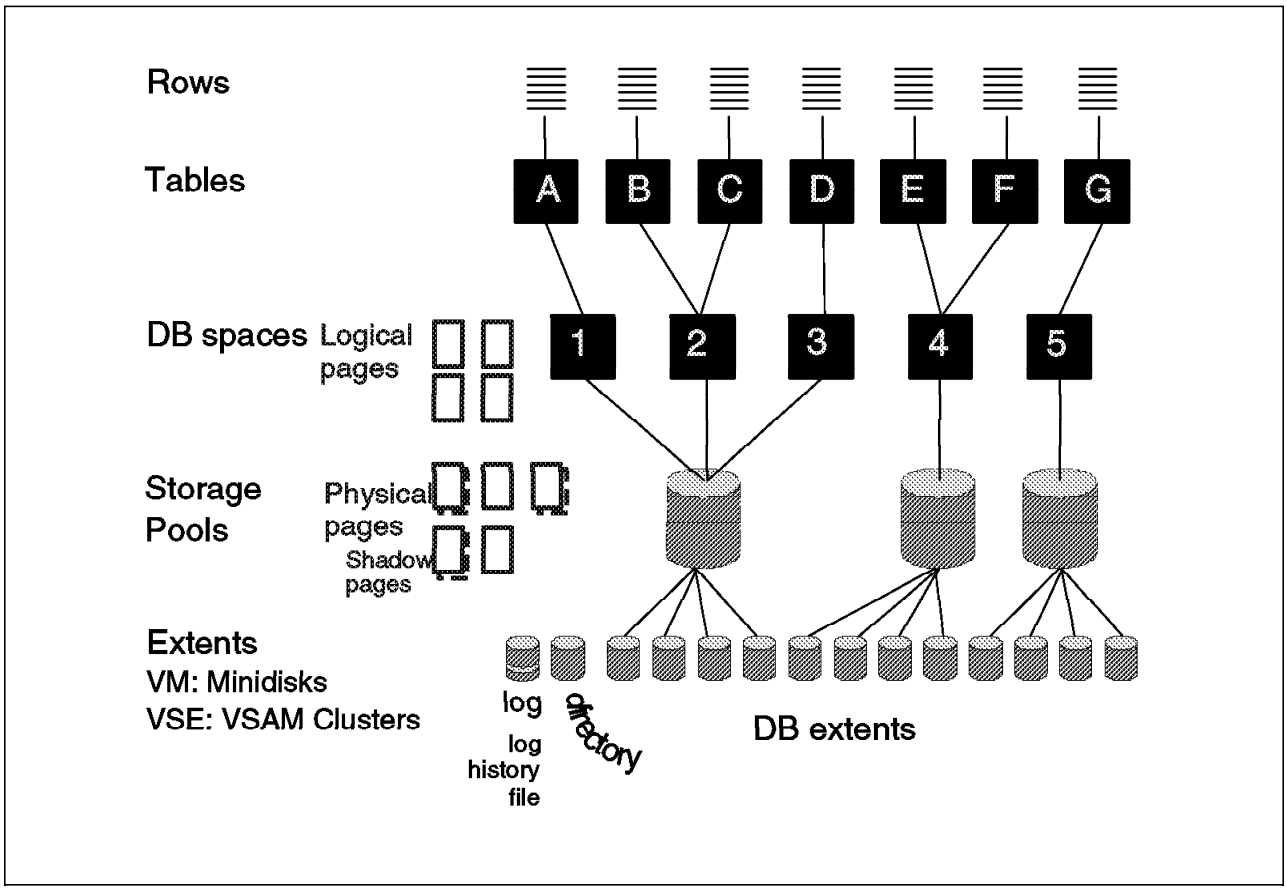


Figure 1. DB2 Structure Overview

The following is a short overview of the DB2 recovery concepts: Data is kept in tables, organized in rows and columns. One or more tables are located in a dbspace. One or more dbspaces are logically stored in a storage pool. A storage pool owns one or more data extents. Other database extents store the database directory and the log. Each extent is represented in VM as a minidisk, in VSE as a VSAM cluster.

- **Physical pages** are owned by the storage pool, logical pages by the dbspace.
- A **logical page** can be represented as two physical pages, a shadow page and a current page.
- **Current pages** are physical pages, containing the actual contents of the database.
- A **shadow page** is a physical page, a copy of a current page, with the original contents at the point in time of the last checkpoint. It is created at the first change after a checkpoint.
- A **checkpoint** frees up all shadow pages. In case of future changes to a logical page, a new shadow page will be created. A checkpoint becomes the starting point for a warm start in case of an abnormal end (abend). Checkpoints occur at startup, shutdown, and other occasions, and are recommended to occur (using a counter) every 15 minutes.

- A **Logical Unit of Work** (LUW), the basic unit of recovery, can either be committed or rolled back, if not all parts of it have been successful. For example, when booking from one account to another, you can join the plus on one and the minus on the other account to become one single entity, so that either both are performed or none.
- The **log** records all changes made to recoverable data. The log is written wrap-around. In LOGMODEs L and A, there are mechanisms to prevent the log from overriding important information.
- The **Log History Area** is the last 4K page of the log. It records recovery events such as archive, log archive, LOGMODE switch, or COLDLOG. This area is vital for recovery.
- **LOGMODE** is a startup parameter. LOGMODE=A allows archives, LOGMODE=L allows archives and log archives. Other LOGMODEs for special purposes are N and Y.
  - LOGMODE=A can automatically start a database archive when a certain threshold value of log usage is reached. Additionally, ARCHIVE can be started manually, which has the advantage that the point in time can be chosen when database activity is low. Before and after the archive, a checkpoint is taken, but LUWs can cross that checkpoint. When the archive has been taken, the log is freed up to the point in time the oldest LUW, crossing the begin archive checkpoint, has started.
  - LOGMODE=L can automatically start a log archive when a certain threshold value of log usage is reached. Additionally, ARCHIVE or LARCHIVE can be started manually, which has the advantage that the point in time can be chosen when database activity is low. Before and after a log archive, a checkpoint is taken. When the log archive has been taken, the log is freed up to the begin log archive checkpoint. If an archive is requested, a log archive is made immediately before. The before log archive checkpoint waits until all LUWs are finished, and new LUWs are prevented until the end log archive checkpoint, or even until the begin archive checkpoint (if ARCHIVE is requested) has finished.

1.3 Functional Overview

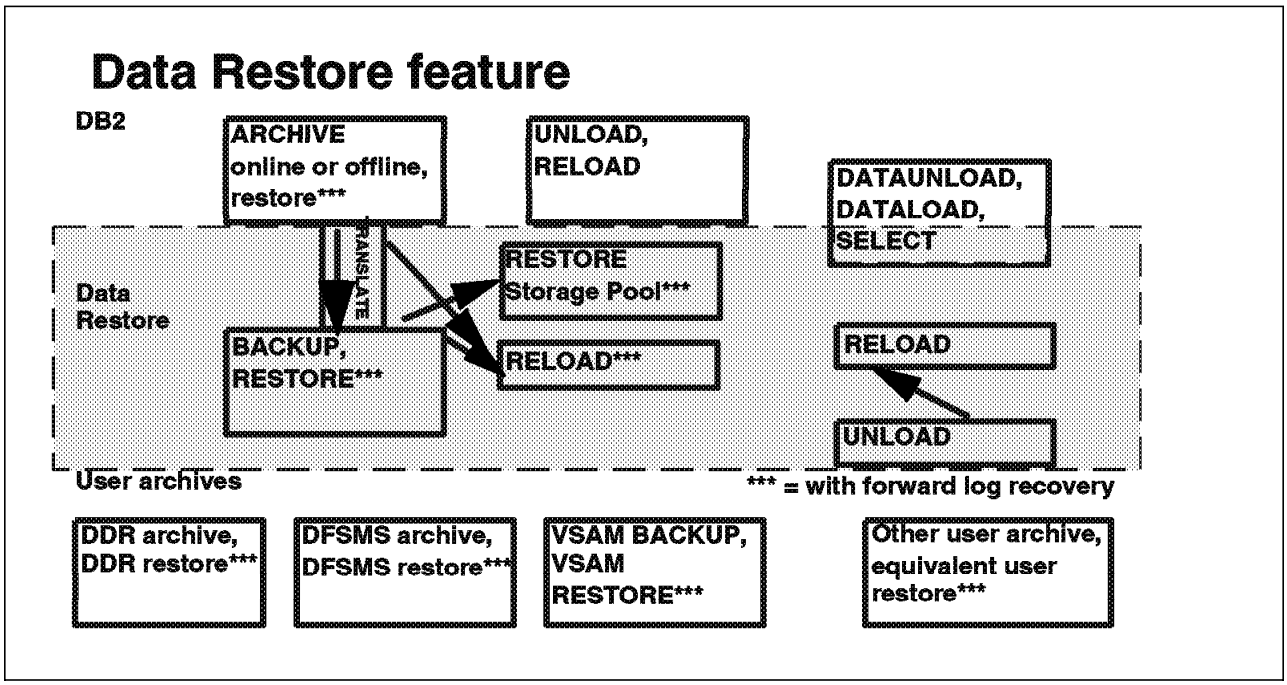


Figure 2. DB2 Recovery Overview

Figure 2 shows all the DB2 recovery capabilities, split into three areas:

1. Inherent DB2 functions (upper area)
2. Data Restore feature functions (medium area)
3. DB2 support for user functions (lower area)

Note: Commands are written in uppercase, functions in lowercase, throughout this book.

The following commands are offered by the Data Restore feature:

BACKUP Back up (Data Restore archive) a complete database to tape or disk with the database manager offline.

RESTORE Restore a storage pool or a complete database from a Data Restore archive.

UNLOAD Selectively back up one or more dbspaces, regardless of whether the database manager is online or offline.

RELOAD Load a table from one of the following:

- a DB2 archive
- a Data Restore archive
- a Data Restore UNLOAD file

and, if option RECOVERY=YES is set, create the necessary work files that allow you to apply log recovery to it. Log recovery is not available on the UNLOAD file. The database manager must be online.

SELECT	Run a query against a table regardless of whether the database manager is online or offline.
FORMAT	Format a dbextent before performing a RESTORE in a VSE environment.
DESCRIBE	List the contents of a Data Restore BACKUP or UNLOAD file to determine which tables can be reloaded. Does not use the database manager; that is, it can be online or offline.
SHOWDBS	Execute a fast SHOW DBSPACE function regardless of whether the database manager is online or offline.
SHOWPOOL	List the storage pool organization by dbextent, and dbextent organization by pool.
LISTLOG	List all the DB2 statements from the log files on disk, written during RELOAD with RECOVERY=YES. This provides the timestamp that can be used later to stop forward recovery at a certain point in time. Does not use the database manager; that is, it can be online or offline.
APPLYLOG	Perform forward recovery: Executes all of the DB2 statements from the work files on disk, written during RELOAD with RECOVERY=YES. The forward recovery can be stopped at any certain point in time. The database manager must be online.
TRANSLATE	Convert a DB2 archive tape to a Data Restore archive (BACKUP) format. Does not use the database manager; that is, it can be online or offline.

VM/VSE Guest Sharing

The Data Restore feature functions are independent of the operating system, VM or VSE. Also the OPTIONS and CONTROL statements apply to both VM and VSE, without any differences. But the Data Restore feature makes use of services of the underlying operating system.

The Data Restore functions can be grouped into four categories:

- TRANSLATE and DESCRIBE are offline functions and do not require any services of the database manager. These functions can run either in VM or VSE and use input tapes originating from both environments.
- RELOAD must be run with the database online. This is the only Data Restore feature function that can be performed in a guest sharing environment, but only if forward recovery is not used. This means that the OPTIONS parameter RECOVERY=NO is specified. The complete description can be found in Chapter 3.5.4, “Data Restore RELOAD” on page 37.
- LISTLOG and APPLYLOG can only be applied on work files that have been created by RELOAD with forward recovery. Therefore, in a VM/VSE Guest Sharing environment, these functions must be applied from VM.
- UNLOAD, SELECT, BACKUP, RESTORE and all other Data Restore functions can or must run with the database manager offline. This implies that the data is extracted directly from the dbextents by means of reading the directory. To access the VM database directory and data disks, CP LINK and CMS ACCESS commands are performed, which cannot be initiated from VSE.

Therefore, databases that are installed under VM and are accessed from VSE through DB2 Guest Sharing must be treated as **VM databases** when using the Data Restore feature functions.

Restrictions

The following restrictions apply to both VM and VSE environments.

- Data Restore feature function RESTORE cannot be run directly with use of a DB2 archive. Before restoring a complete database with the Data Restore feature, the archive must be translated with Data Restore TRANSLATE, or the database can be restored using the standard DB2 restore facility.
- Data Restore RELOAD cannot reload the system catalog tables.
- For the Data Restore SELECT, any table can be specified that does not contain LONG columns.
- The Data Restore SELECT function neither executes local date and time user exits, nor does it execute field procedures on the selected columns.

1.4 Command Syntax

The commands for the module or phase XTS91001 are specified through a SYSIN file in VM, or directly following the EXEC XTS91001 JCL statement in VSE. In VM there must be a FILEDEF SYSIN that points to the file that contains the input commands.

Following are some remarks about the command syntax.

General Remarks for VM and VSE:

- All statements, such as OPTIONS and CONTROL, and all commands, such as RESTORE and BACKUP, must start in column 1. To continue a statement or command on different lines, just add the continuing lines starting from column 2.
- Comments are identified by an asterisk in column 1.
- The order for the statements is OPTIONS, CONTROL (if needed), then followed by other commands such as RELOAD, BACKUP.
- Upper and lower case can be mixed in the same statement or command.
- Parameters, such as ARCHTYPE and DEVICE, can be separated by a space or a comma in the same statement or command.
- If Data Restore feature functions are being used in unattended procedures and you want to avoid the prompt, indicated by message XTS9-406, requesting a reply for continue or cancel, add the OPTIONS statement CONFIRM=NO to your input definitions.

Specific Remarks:

VSE and VM

- OPTIONS DEVICE=DASD specifies the input or output device for ARCHIV and must be specified if a file on disk is used.
- Specify CONTROL DBNAME=dbname, although this is assumed to be optional.

- Ddname ARCHIV points to the output file of BACKUP, or to a workfile (during TRANSLATE or RELOAD from DB2 archive).
- Ddname ARIARCH points to a DB2 archive tape.
- Ddname LARCHIV points to the log archive tapes or files. (ARILARC must not be used in Data Restore feature Version 3 Release 5 but is supported by Data Restore feature Version 5 Release 1.)

VSE

- SYS006 is always related to ARCHIV (Data Restore archive), both for input and output.
- SYS007 is always related to ARIARCH (DB2 archive) input.
- For the RELOAD function with forward recovery and LOGMODE=L, the log archive tapes must be mounted on the same device as the tapes pointed to by TLBL ARCHIV. The assignment for this device is SYS006.
- To avoid the messages 4228I and 4233I (during a follow-on run of TRANSLATE or RELOAD) indicating that the file already exists, the SYS0001, HEADER and DIRWORK files can be defined with a retention period of zero days. Refer to Figure 66 on page 92 for a sample definition of these files.
- To be able to RELOAD from a translated DB2 archive file, you must keep a copy of the SYS0001, HEADER and DIRWORK files which were created at the time the TRANSLATE was performed. This means, a retention period of 0 days is **not** suitable. After having saved them somewhere else, they must explicitly be deleted before a follow-on run of TRANSLATE or RELOAD.

VM

- For the RELOAD function with forward recovery, in the filedef statements, the device name TAP1 must point to the log archive tape and TAP2 must point to the archive tape from DB2 archive or Data Restore archive.
- For the ddname ARCHIV, the record format must always be specified as variable blocked (RECFM VB) and blocksize 32760 (BLOCK/BLKSIZE 32760).

Chapter 2. Archive and Recovery

This chapter describes the options provided to back up a complete DB2 database and to restore this data.

Please refer to Appendix A, “Test Scenarios Using SQL/DS V3R5” on page 73 and Appendix B, “Test Scenarios Using DB2 Server for VSE & VM V5R1” on page 79 which provide our test results. These give you an idea of how much time each process took in a test environment. The figures can help you to decide when or how often these processes should be performed. These are directly related to the amount of active pages in the database and the size of the directory disk.

Also the specifications for the work files can be found in Figure 66 on page 92.

2.1 Overview

With the Data Restore feature, there are additional possibilities to manage the backup and recovery process.

The different possibilities explained are:

- User archives with non-DB2 tools
- DB2 ARCHIVE and restore
- Data Restore BACKUP and RESTORE

2.1.1 Control Center

Control Center is a new DB2 Server for VSE & VM V5R1 feature and replaces the product SQL Master. As a DB2 feature, it supports the use of Data Restore feature under VM environment. With this support, you can explore the new Data Restore feature functions which facilitate database maintenance for the administrator.

Control Center, can help you automate the process for taking archives. Control Center can schedule an automated archive at predefined times or when certain conditions are met.

Control Center has also some interfaces that let you take non-DB2 archives or initiate backups using other user or vendor written programs, for example VMBACKUP or VMTAPE.

For a complete example of Control Center using Data Restore TRANSLATE function, please refer to C.7, “Example of Data Restore TRANSLATE Using Control Center” on page 97.

For a detailed description of archive and/or backup automation refer to *Control Center for VM Installation and Operations Guide*.

2.2 User Archives

There are many ways to perform a backup depending on the operating system and on the programs installed in this environment.

An archive taken using non-DB2 facilities is called a *user archive.* It can only be performed offline, that is when the database manager has been shut down by SQLEND UARCHIVE. Never use SQLEND or SQLEND QUICK when performing a user archive. You must use SQLEND UARCHIVE, because this:

- Creates an entry in the log history area.
- Makes a checkpoint, and thus leaves the database in a clean status for the archive.
- Avoids unnecessary UNDO/REDO operations after restoring the database from a user archive.

User archives let you use a variety of tools and you can switch between different types of archive as often as you like. The following is a list of non-DB2 archive tools available:

- DDR

  Available on VM only. It can be executed in VM as a module or from a standalone tape to back up either a VM minidisk or a DASD belonging to VSE.

  The disadvantage of DDR is that it requires some input definitions to be supplied and these must be maintained in case the physical layout of the database changes.

  In VM this means, that the procedure that initiates the copy needs to know about all the minidisks belonging to the database. Use the current directory file, for example USER DIRECT, or the SQLFDEF file that is maintained by the database manager and is available on the DB2 production disk.

  In VSE, only complete disks, not dbextents, can be copied, and VSAM definitions on those disks must not be changed.

- DFSMS

  Available on VM only.
  It performs better than DDR in most cases but, as DDR, it requires that you supply input definitions and maintain them in case the physical layout of the database changes.

- VSAM IDCAMS Backup/Restore

  Available on VSE only. IDCAMS BACKUP can be applied. As with DDR and DFSMS, VSAM Backup/Restore requires some input parameters related to the extents and must be maintained if the design of the database changes. For a detailed description of this command see the *IBM VSE/VSAM Commands* manual.

  The VSAM IDCAMS REPRO command can **not** be used for the backup of DB2 extents.

- Any other user or vendor written application that can perform any type of backup of complete disks or minidisks. Most of these backup applications are based on DDR in VM for the copy of a whole minidisk.

Restore is made using the same tool as the archive, and then restarting the database with STARTUP=U for forward log recovery.

Note: Typically, no table level recovery is available using these user archive tools.

2.3 Archiving with DB2

There are two kinds of archives using the DB2 product; a database archive and a log archive.

2.3.1 DB2 Archive

A database archive (DB2 archive) is a tape copy of the database directory and the dbextents. The database manager takes a checkpoint (the begin-archive checkpoint) and writes a copy of the database directory and the active data pages to tape, as they were at the checkpoint. A database archive does not include a copy of the log.

Archives can be taken either online (ARCHIVE) or in the process of being shut down, called “offline” (SQLEND ARCHIVE).

- An **offline** archive is consistent in itself because there are no ongoing updates. On restore, log recovery can be applied.
- An **online** archive
  - with LOGMODE=A, makes a checkpoint before and after the archive. LUWs can exist during the begin archive checkpoint. Therefore, the archive can be inconsistent and needs forward recovery through the current log to become consistent when restored.
  - with LOGMODE=L, enforces a log archive before, and checkpoints before and after the log archive and the archive. The begin log archive checkpoint waits for the end of any active LUWs and new LUWs are prevented until the end of the begin archive checkpoint. Therefore, the archive is consistent in itself.

In SQL/DS V3R5, enhancements were made to improve the performance of the archive process:

- The database manager archive process only archives pages that have been allocated. Non-allocated pages are not archived. This not only reduces the time but also the number of tapes needed to back up the database.
- In **VM**, it uses asynchronous I/O to reduce the wait time for reading DASD by having concurrent disk and tape operations.
- In **VM**, it uses Multiple-Block *BLOCKIO requests in a single IUCV SEND request.
- In **VSE**, it uses VSAM controlled buffers. By doing this, VSAM is able to read multiple records with a single I/O request. This improvement in VSE is equivalent to both VM improvements above, asynchronous I/O and multiple block *BLOCKIO.

A detailed description and a test result overview can be found in the *SQL/DS Version 3 Release 5 Usage Guide*.

2.3.2 Log Archive

A log archive is a tape or disk copy of the log, recording just the changes applied to the data since the last archive or log archive.

Log archives can be made between DB2 archives or user archives to improve the availability of the database. In most situations, if you are using log archive, the period between database archives is increased.

Warning: If you use LOGMODE=L on a database with lots of updates, when you restore the database the log recovery will have to re-execute all changes. This process can be very time consuming and use a lot of resources from the database or the system, especially the processor (CPU).

Only DB2 facilities can be used to archive the log. The database manager must be running with the startup parameter LOGMODE=L. Log archives are taken when the database manager is either running or still running but in the process of being shut down.

The output for the log archive is defined by the FILEDEF (in VM) or TLBL/DLBL (in VSE) statement with ddname ARILARC. To be able to restore a database to its current level the log archives must be continuous.
For filtered log recovery, which excludes some operations from the UNDO/REDO process, refer to the *DB2 Diagnosis and Reference* manual.
For a detailed description of log archives refer to *DB2 System Administration: Performing Log Archives*.

2.4 DB2 Restore

To perform the recovery procedure, the database manager has to be started with the STARTUP=R parameter. The directory and all dbextents are reloaded from the DB2 archive tape. Log archive files and log recovery can be applied after the restore of the data.
In VSE, STARTUP=F is equivalent to STARTUP=R in VM; in VSE, STARTUP=R first formats all dbextents, which can take longer than the restore process itself.

If LOGMODE=A and the archive being restored is not the last one taken, you must do a COLDLOG to avoid inconsistency or an abnormal end. But even so, if the archive restored was an online archive, the inconsistency due to active LUWs during the online archive cannot be resolved.

If LOGMODE=A and the latest archive is restored, the current log can be used for the UNDO/REDO process. This process makes a online archive, taken with LOGMODE=A, become consistent.

If LOGMODE=L, do not perform COLDLOG because this would prevent you from applying the log archives and the current log.

For detailed information how to restore a database from a DB2 archive refer to *DB2 System Administration : Recovery Procedures*.

2.5 Log Recovery

- At database startup after restoring a user archive (STARTUP=U) or when restoring a DB2 archive (STARTUP=R or STARTUP=F in VSE), **forward log recovery** is applied. This includes log archives, if LOGMODE=L. The log archives must be without a gap. The log recovery is stopped as soon as one file is missing.

  You can **filter** your log for the UNDO/REDO process. This means you can exclude some specific DB2 command(s) from the recovery process, and after that specific command the log application is resumed.

- With Data Restore RELOAD, LISTLOG and APPLYLOG you can **apply log recovery** (to the tables restored) **up to a certain point in time**. But unlike filtered log recovery, the log recovery cannot be resumed if it stops because it has:

  – Reached a certain predefined point in time

  – Encountered a DROP TABLE, ALTER TABLE or DROP DBSPACE command

  During RELOAD you are prompted for the log archives. You are prompted if you want to skip a log archive tape or file. The LISTLOG and APPLYLOG will execute only on the DB2 commands loaded into the work files during RELOAD.

- If you are using LOGMODE=L, after a Data Restore **RESTORE of a storage pool from an older archive** than the most recent, you have to apply all the log archives in between, including the current log, in order to have the information up to date and consistent. If you are running in LOGMODE=A, you have to either restore a full archive, or restore a storage pool from the most recent archive.

2.6 Data Restore BACKUP

The Data Restore BACKUP command creates a Data Restore archive of the entire database. The Data Restore BACKUP command is performed as a DB2 *user archive*, thus the database has to be offline. As the DB2 ARCHIVE, it includes the directory and dbextents and does not include the log.

The output can be used as input for either:

- the Data Restore command RESTORE to restore a storage pool
- the Data Restore command RESTORE to restore the whole database
- the Data Restore command RELOAD for a selective restore of tables

Data Restore BACKUP writes additional information to the beginning of the archive file which is used for a later Data Restore RELOAD of tables. This is equivalent to the work files SYS0001, HEADER and DIRWORK of TRANSLATE, as described in chapter 2.7, “Data Restore TRANSLATE” on page 14.

If recovery to a certain ′point in time′ is needed, the DB2 database must be running with LOGMODE=L or LOGMODE=A. This means that successive logs can be applied up to a certain point in time.

Data Restore BACKUP can be directed to tape, disk or both.

Files

The Data Restore BACKUP requires some input and output files:

For **VM** only:

- **SYSIN** file - the file contains the command to be processed.
- **SYSPRINT** file - Data Restore feature creates a report that lists the SYSIN values, messages and results.

For **VM** and **VSE**:

- **ARCHIV** - first copy
- **ARCHIV2** - second copy

If only one output is needed the FILEDEF (in VM) and DLBL/TLBL (in VSE) for ddname ARCHIV and the OPTIONS DEVICE= TAPE/DASD defines this output to tape or disk.

If dual backup is needed then the FILEDEF (in VM) or DLBL/TLBL (in VSE) for ddname ARCHIV2 and the OPTIONS DEVICE2=DASD or TAPE define the second output, which can be again either to tape or disk. Refer to Figure 55 on page 87 for a sample JCL, or Figure 56 on page 87 and Figure 57 on page 87 for a sample exec and SYSIN for the BACKUP.

In **VM** the user, running the backup, must have access to the database production disk, default 195, because the "*dbname* SQLFDEF" file is used to link and access the database minidisks.

In **VSE**, the backup job must also execute the DLBL statements for directory and dbextents.

For backup **to disk**, the following restrictions apply:

In **VM**, the size of the output file is limited to the size of the minidisk, and because a minidisk can only be defined on a single volume, the maximum size is the size of the volume.

In **VSE**, the space for this file can be allocated on different volumes, by defining space for this user catalog on different volumes. The size of the file is limited to 4GB by VSAM, and to the physical space that is available for allocation by this catalog.

2.7 Data Restore TRANSLATE

The Data Restore feature command TRANSLATE can be used to increase the performance when later restoring the complete database or parts of it from a DB2 archive. TRANSLATE reads the DB2 archive tapes twice and creates tapes or disk files that can be used as input for the Data Restore RESTORE or RELOAD, as if it were a Data Restore archive made via Data Restore BACKUP. TRANSLATE improves the performance of the recovery operation.

TRANSLATE is required to Data Restore RESTORE from a DB2 database, or the database can be restored using the standard DB2 restore facility. For Data Restore RELOAD, TRANSLATE is optional.

The TRANSLATE process can be started at any time and does not affect the operation of your database manager. When this process should run depends on your environment:

- either immediately following the archive,
- or preceding the restore of either the database (RESTORE) or part of the database (RESTORE, RELOAD).

Files

The Data Restore TRANSLATE requires some input and output files:

For **VM** only:

- **SYSIN** file - contains the command to be processed.
- **SYSPRINT** file - Data Restore feature creates a report that lists the SYSIN values, messages and results.

For **VM** and **VSE**:

- **ARCHIV** - workfile
- **ARIARCH** - input file from a DB2 archive
- **LMBRWRK** - workfile or cluster which contains pages for tables containing LONG columns.

SYS0001, HEADER and DIRWORK will be used by a later RELOAD function.

- **SYS0001** - workfile, stores active pages
- **HEADER** - workfile, stores header pages
- **DIRWORK** - workfile, stores the directory pages

Example

Refer to Figure 3 for a sample command syntax that was used to translate a DB2 archive to a Data Restore archive format.

```
* $$ JOB JNM=DRFTRANS,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFTRANS              TRANSLATE SQL ARCHIVE TO DRF BACKUP
// LIBDEF *,SEARCH=(PRD2.SQL350,PRD2.RCV350)
// TLBL     ARIARCH
// TLBL     ARCHIV
// ASSGN    SYS007,180             input tape : SQL/DS archive
// ASSGN    SYS006,181             output tape
// MTC      REW,SYS007
// MTC      REW,SYS006
// DLBL DIRWORK,,,VSAM,CAT=VSESPUC
// DLBL SYS0001,,,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL HEADER,,,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// EXEC XTS91001,SIZE=AUTO
TRANSLATE
/*
/&
* $$ EOJ
```

Figure 3. JCL File for Data Restore TRANSLATE

The Data Restore TRANSLATE produces the work files, SYS0001, HEADER and DIRWORK. If you plan to RELOAD from a translated archive file, you must keep a copy of these work files in both VM and VSE. If you plan to RESTORE, these files can be destroyed after the TRANSLATE. In this case it is recommended in VSE to define these files with a retention period of zero days in order to avoid in the case of a follow-on run, the messages *4228I* and *4233I*, indicating that the "file already exists."

2.8 Data Restore DESCRIBE

To Data Restore RESTORE a Data Restore archive that was not the last archive taken, you first have to run the Data Restore DESCRIBE function in order to have the correct identification of that archive. It also gives you a list of tables that are available in this archive file and can be reloaded individually if necessary.

The DESCRIBE function can be run against any of the following:

- A Data Restore archive (the output of a Data Restore BACKUP)
- A translated DB2 archive
- A Data Restore UNLOAD

The archive identification must be specified in the CONTROL statement with "DATE=" and "TIME=" for the Data Restore RESTORE. Refer to Figure 5 on page 17 for a sample CONTROL statement that defines an archive to be restored.

Files

The Data Restore DESCRIBE requires some input and output files:

For **VM** only:

- **SYSIN** file - the file contains the command to be processed.
- **SYSPRINT** file - Data Restore feature creates a report that lists the SYSIN values, messages and results.

For **VM** and **VSE**:

- **ARCHIV** - input file to be described

Example

Figure 63 on page 90 shows the exec used for this function in VM, and Figure 4 on page 17 shows a VSE output sample of DESCRIBE.

```
// JOB DRFDESCR               DESCRIBE ARCHIVE TAPE
// LIBDEF *,SEARCH=(PRD2.SQL350,PRD2.RCV350)
// TLBL     ARCHIV             ** input file **
// ASSGN    SYS006,181
// MTC      REW,SYS006
// EXEC XTS91001,SIZE=AUTO
CONTROL DBNAME=SQLVSE02
DESCRIBE

XTS9-143 CONTROL DBNAME=SQLVSE02
XTS9-143 DESCRIBE
XTS9-143 /*
XTS9-100 Data Restore feature 3.5.0
XTS9-192 Processing SQL/VSE  archived by SQL/DS on (05/21/96-07-47-03)
XTS9-013 Table SQLDBA  .ACTIVITY          may be reloaded
XTS9-013 Table DATARFTR.CMD               may be reloaded
XTS9-013 Table SQLDBA  .CUSTOMER          may be reloaded
 ... more entries in between .........
XTS9-013 Table SQLDBA  .WAREHOUSE         may be reloaded
XTS9-007 Processing successfully completed
```

Figure 4. SYSPRINT File of Data Restore DESCRIBE

```
CONTROL DATE=05/21/96 TIME=07-47-03
        BASE=S35VMDB1
RESTORE
```

Figure 5. CONTROL Statement Defining an Archive for Data Restore RESTORE

2.9 Data Restore FORMAT

In **VSE**, when a dbextent is damaged or has to be moved to a different volume, the extent has to be defined using the IDCAMS DEFINE CLUSTER command. Before restoring the storage pool or the database using the Data Restore RESTORE command, you must prepare the defined cluster for DB2. This process only applies for data extents.

It does **not** apply for:

- the directory disk (BDISK). The Data Restore feature formats the BDISK while restoring the database.
- the log disk (LOGDSK1). To format the log disk, the COLDLOG procedure with the parameters SYSMODE=S and STARTUP=L should be used.

If in VSE DB2 facilities are used to restore a database, you have two choices:

- The Data Restore FORMAT is not needed, as the DB2 restore procedure with STARTUP=R first formats all dbextents and then restores the database.
- You can use Data Restore FORMAT to format just one extent, for example after replacing, and then you can apply the DB2 restore procedure with STARTUP=F, so that it omits formatting. This can be much quicker, depending on the number of extents used.

In **VM**, the CMS commands FORMAT and RESERVE can be used to perform the equivalent function for both follow-on processes, Data Restore RESTORE and DB2 restore.

Example

Figure 6 shows a sample of the JCL to format the extent, known as DDSK10 by DB2.

```
* $$ JOB JNM=DRFORMAT,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFORMAT                format data extent
// LIBDEF *,SEARCH=(PRD2.SQL350,PRD2.RCV350)
// EXEC PROC=SQLVSE02
// EXEC XTS91001,SIZE=AUTO
OPTIONS RECOVERY=NO,DEVICE=DASD
CONTROL DBNAME=SQLVSE02
FORMAT DDSK=10
/*
/&
* $$ EOJ
```

Figure 6. JCL File for Data Restore FORMAT Dbextent

2.10 Data Restore SHOWDBS

The Data Restore SHOWDBS executes a SHOW DBSPACE function while the database is online or offline. The example below lists all dbspaces in the database with a summary.

```
XTS9-143 OPTIONS CONFIRM=NO
XTS9-143 CONTROL BASE=S35VMDB1
XTS9-143 SHOWDBS
XTS9-143 /*
XTS9-100 Data Restore feature 3.5.0
XTS9-313 SQL/DS Version 3 processed
XTS9-601 DATE:17/06/96  TIME:15:02:28
------------------------------------------------------------------------    -------
XTS9-014 Pool   1 Dbno   1 Name SYS0001           Pages    12.800 Header 8 Index 60

XTS9-019          Header       Data      Index
 XTS9-020         ------  ---------  ---------
XTS9-018 Maximum       8      5.112      7.680
XTS9-015 Reserved      1         87         54
XTS9-016 Used          1         86         54
XTS9-017 %Used      12,5        1,6        0,7
-------------------------------------------------------------------------------------
XTS9-014 Pool   1 Dbno   2 Name SYS0002           Pages     2.048 Header 8 Index  0

XTS9-019          Header       Data      Index
 XTS9-020         ------  ---------  ---------
XTS9-018 Maximum       8      2.040          0
XTS9-015 Reserved      6         55          0
XTS9-016 Used          6         53          0
XTS9-017 %Used      75,0        2,5        0,0
-------------------------------------------------------------------------------------
  .. more pools in between ....
-------------------------------------------------------------------------------------
XTS9-601 DATE:17/06/96  TIME:15:02:28
XTS9-022 Pool:001
XTS9-023 ========
XTS9-024         Dbno Dbspacename         Reserved      Used     Empty
XTS9-025         ---- ----------------- --------- --------- ---------
XTS9-021            5 SAMPLE                   20        12         8
XTS9-021            2 SYS0002                  61        59         2
XTS9-021            4 ISQL                      8         7         1
XTS9-021            1 SYS0001                 142       141         1
XTS9-026                                  --------- --------- ---------
XTS9-027 Total pool:  1                       231       219        12
XTS9-601 DATE:17/06/96  TIME:15:02:28
XTS9-007 Processing successfully completed
```

Figure 7. SYSPRINT File of Data Restore SHOWDBS

2.11 Data Restore SHOWPOOL

The Data Restore SHOWPOOL executes a SHOW POOL function while the database is online or offline. This function is used to identify which storage pool should be restored by listing the dbextents associated with each storage pool defined on the database server. The example below lists all storage pools in the database.

```
XTS9-143 CONTROL DBNAME=SQLVSE02
XTS9-143 SHOWPOOL
XTS9-143 /*
XTS9-100 Data restore feature version 4.1.0
XTS9-313 DB2 for VSE and VM Version 4 processed
XTS9-143 ************************************************************
XTS9-203 Pool=   1 First dbextent=    1    Pool is RECOVERABLE
XTS9-203 Pool=   2 First dbextent=    2    Pool is RECOVERABLE
XTS9-203 Pool=   3 First dbextent=    3    Pool is RECOVERABLE
XTS9-203 Pool=   4 First dbextent=    4    Pool is RECOVERABLE
XTS9-203 Pool=   5 First dbextent=    5    Pool is RECOVERABLE
XTS9-203 Pool=   6 First dbextent=    6    Pool is RECOVERABLE
XTS9-203 Pool=   7 First dbextent=    7    Pool is RECOVERABLE
XTS9-203 Pool=   8 First dbextent=    8    Pool is RECOVERABLE
XTS9-203 Pool=   9 First dbextent=    9    Pool is RECOVERABLE
XTS9-203 Pool=  10 First dbextent=   10    Pool is RECOVERABLE
XTS9-203 Pool=  11 First dbextent=   11    Pool is RECOVERABLE
XTS9-203 Pool=  12 First dbextent=   12    Pool is RECOVERABLE
XTS9-203 Pool=  13 First dbextent=   13    Pool is RECOVERABLE
XTS9-203 Pool=  14 First dbextent=   14    Pool is RECOVERABLE
XTS9-143 ************************************************************
XTS9-204 Dbextent=    1 Pool=  1 Next dbextent=NONE
XTS9-204 Dbextent=    2 Pool=  2 Next dbextent=NONE
XTS9-204 Dbextent=    3 Pool=  3 Next dbextent=NONE
XTS9-204 Dbextent=    4 Pool=  4 Next dbextent=NONE
XTS9-204 Dbextent=    5 Pool=  5 Next dbextent=NONE
XTS9-204 Dbextent=    6 Pool=  6 Next dbextent=NONE
XTS9-204 Dbextent=    7 Pool=  7 Next dbextent=NONE
XTS9-204 Dbextent=    8 Pool=  8 Next dbextent=NONE
XTS9-204 Dbextent=    9 Pool=  9 Next dbextent=NONE
XTS9-204 Dbextent=   10 Pool= 10 Next dbextent=NONE
XTS9-204 Dbextent=   11 Pool= 11 Next dbextent=NONE
XTS9-204 Dbextent=   12 Pool= 12 Next dbextent=NONE
XTS9-204 Dbextent=   13 Pool= 13 Next dbextent=NONE
XTS9-204 Dbextent=   14 Pool= 14 Next dbextent=NONE
XTS9-143 ************************************************************
XTS9-007 Processing successfully completed
```

Figure 8. SYSPRINT File of Data Restore SHOWPOOL

2.12 Data Restore RESTORE

With DB2 Server for VSE & VM V5R1, a new function which provides storage pool level recovery was implemented to the Data Restore RESTORE command.

Now the Data Restore RESTORE command can be used to recover a single storage pool, a set of storage pools or an entire database after a system or disk failure, or to create a copy of the entire database on the same or another system. To run Data Restore RESTORE, the database manager must be offline.

For information about log recovery after RESTORE of a storage pool, please refer to 2.5, “Log Recovery” on page 13.

The input can be any of the following:

- A Data Restore archive (the output of a Data Restore BACKUP).
- A translated DB2 archive.

In **VSE**, the Data Restore feature formats the BDISK while restoring the database.

In **VM**, if the Data Restore feature attempts to read or write the database BDISK, a dbextent or log disk, the minidisk will be linked as the same CUU as the original database minidisk. The Data Restore feature must have authority to link to all database minidisks in write (W) mode. If a minidisk needs to be accessed, it will be accessed as filemode B, C, D, or E. These filemodes should be available before invoking RESTORE.

Files

The Data Restore RESTORE requires some input and output files:

For **VM** only:

- **SYSIN** file - the file contains the command to be processed.
- **SYSPRINT** file - Data Restore feature creates a report that lists the SYSIN values, messages and results.

For **VM** and **VSE**:

- **ARCHIV** - Data Restore archive file to be restored

Example

Figure 9 shows a sample of the JCL to use Data Restore RESTORE.

```
* $$ JOB JNM=DRFRESTR,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFRESTR              RESTORE DATABASE WITH DRF from tape
// LIBDEF *,SEARCH=(PRD2.SQL350,PRD2.RCV350)
// EXEC PROC=SQLVSE02
// EXEC PROC=XTS9DLBL
// TLBL     ARCHIV
// ASSGN    SYS006,181
// MTC      REW,SYS006
/* PAUSE for monitor
// EXEC XTS91001,SIZE=AUTO
CONTROL DBNAME=SQLVSE02
RESTORE
/*
/&
* $$ EOJ
```

Figure 9. JCL File for Data Restore RESTORE

Figure 10 on page 22 shows a sample of the EXEC to use Data Restore RESTORE.

```
/* */
/* FILEDEF FOR INPUT FROM TAPE */
¢FILEDEF ARCHIV   TAP1 SL (RECFM VB BLOCK 32760¢
¢FILEDEF SYSIN    DISK RESTORE SYSIN A¢
¢FILEDEF SYSPRINT DISK RESTORE SYSPRINT A¢
¢XTS91001¢
Exit rc
```

Figure 10. EXEC File for Data Restore RESTORE from Data Restore BACKUP (RESTORE EXEC)

Figure 11 shows a sample of the EXEC to use Data Restore RESTORE.

```
OPTIONS DEVICE=TAPE CONFIRM=NO
CONTROL  BASE=ELDB2A
RESTORE
```

Figure 11. SYSIN File for Data Restore RESTORE from Data Restore BACKUP (RESTORE SYSIN)

2.13 Summary

2.13.1 Compatibilities

Data Restore BACKUP uses a different format than DB2 ARCHIVE. Therefore, a Data Restore archive cannot be used for a DB2 restore.

However, you can change a DB2 archive to a Data Restore archive format using the Data Restore TRANSLATE command; the only difference is that after TRANSLATE the work files need to be kept separately for a future Data Restore RELOAD with forward recovery, whereas this information is already at the beginning of a Data Restore archive file after BACKUP.

Table 1 gives you an overview of the compatibilities among the methods that can be used to archive and recover your DB2 database.

Table 1. Compatibilities Archive and Recovery

	... can be **Input** to:					
Output from ...	DB2 RESTORE	Data Restore TRANSLATE	Data Restore DESCRIBE	Data Restore RELOAD	Data Restore RESTORE	Equivalent user restore
DB2 archive	**X**	**X**	-	**X**	-	-
Translated DB2 archive	-	-	**X**	**X**	**X**	-
Data Restore UNLOAD	-	-	**X**	**X**	-	-
Data Restore BACKUP	-	-	**X**	**X**	**X**	-
Any user archive	-	-	-	-	-	**X**

2.13.2 Possibilities

The DESCRIBE function of the Data Restore feature lists the contents of a Data Restore BACKUP or UNLOAD file to determine which tables can be reloaded.

For the difference between filtered log recovery and APPLYLOG with forward log recovery to a certain point in time, see chapter 2.5, “Log Recovery” on page 13.

The following table gives you an overview of the methods and their possibilities to archive your DB2 database.

Table 2. Possibilities to Archive

		SQL/DS	**Data Restore**
Function	User archive	ARCHIVE	BACKUP
Status database			
online	-	**X**	-
offline	**X**	**X**	**X**

Table 3 gives you an overview of the methods and their possibilities to restore your DB2 database. For “Data Restore RELOAD from TRANSLATE” (translated DB2 archive) see column “RELOAD from BACKUP”

Assumption: The database is running with LOGMODE=L or LOGMODE=A.

Table 3. Possibilities to Recover

		SQL/DS	**Data Restore feature**			
Function	restore from User archive	restore from ARCHIVE	RESTORE	RELOAD from BACKUP	RELOAD from ARCHIVE	RELOAD from UNLOAD
Unit						
table	-	-	-	**X**	**X**	**X**
storage pool	-	-	**X**	-	-	-
database	**X**	**X**	**X**	-	-	-
Status database						
online	-	-	-	**X**	**X**	**X**
offline	**X**	**X**	**X**	-	-	-
Restore						
last archive	**X**	**X**	**X**	**X**	**X**	-
any archive	**X**	**X**	**X**	**X**	**X**	-
Recover log						
apply all	**X**	**X**	**X**	**X**	**X**	-
to log file border	**X**	**X**	**X**	-	-	-
to a point in time	-	-	-	**X**	**X**	-
filtered log	**X**	**X**	**X**	-	-	-

Chapter 3. Data Unload and Reload

This chapter describes the options provided to back up selected data in different formats in order to load it back to a relational database management system or other database or application on the same or on a different operating system.

Likewise, it describes how to load data from various formats, including a partial or complete database backup. With user defined formats, the sources of the data can be any database or application on any platform.

Please refer to Appendix A, “Test Scenarios Using SQL/DS V3R5” on page 73 which provides our test results. These give you an idea about how much time each process took in a test environment. The figures can help you to decide when or how often these processes should be performed. They are directly related to the amount of active pages in the database and the size of the directory disk.

3.1 Overview

Unload and load procedures are used when you want to reorganize the data, import or export data, or back up or restore part of the database.

If you used unload procedures previously, you could not apply a log. That is, you could not automatically track and apply any changes that were made to the data after the unload was taken. However, there is now the possibility to RELOAD a specific table from a Data Restore archive or DB2 archive and apply the log.

The main reasons to unload data are to:

- Reorganize a table
  - when the indexes become unclustered
  - when the table specifications need changes
  - when the dbspace specifications need changes
- Copy a complete table or a part of it from one database to another to create a test environment.
- Use the unloaded data as input to another relational database or even to a non-relational database, for example a spreadsheet or reporting application.
- Back up critical tables between archives or log archives:
  - If something goes wrong with your database, it would be faster to restart the environment without having to restore all data.
  - If the database manager cannot be restarted at all, you would at least have a recent copy of important tables.

  But note that no log can be applied and thus changes to those tables are not reflected. These table backups are now no longer necessary when using the Data Restore feature.

3.2 DBSU DATAUNLOAD and DATALOAD

DATAUNLOAD and DATALOAD are part of the DBSU.

DATAUNLOAD enables you to unload data from DB2 tables to a file with a user-defined format. The data to be unloaded is selected from the database within an SQL SELECT statement. The DATAUNLOAD command can have some subcommands that describe the data fields and the source in the output records. Refer to Figure 12 on page 27 and to Figure 13 on page 27 for a sample of the DATAUNLOAD command sequence. In most cases, each output record contains data from one row of a table. The record format can vary and can have different formats not only depending on the operating system, but also based on the user-defined record layout.

DATALOAD allows you to reload rows into existing DB2 tables from data contained in a sequential input file that was created by either DATAUNLOAD or a process outside the DB2 system.

For a complete and detailed description of these commands, refer to the *DB2 Database Services Utility* manual.

Purpose

Some of the different purposes of this facility are:

Reorganization

Indexes can become unclustered when rows in a table are being added, deleted or updated. So there may be a need to unload and reload the data to cluster the indexes again. Reorganization is also useful if many rows have been deleted and the number of empty pages is too high.

If reorganization is the main purpose, it is easier and faster to use the method described in chapter 3.3, “DBSU UNLOAD and RELOAD” on page 28.

To unload data in a specific order and reload it independent of the clustering index you can use ORDER BY on the DATAUNLOAD command.

Restructure

A change in the structure of a table may be required. If the layout of the table, the datatype of the columns or the NULL definition needs changing, consider using DATAUNLOAD and DATALOAD because you can provide the table definition with the DATALOAD command sequence. DATAUNLOAD and DATALOAD would be the best way to do this.

DATAUNLOAD should also be used if columns need the NOT NULL specification, or columns have to be created that would be parts of an existing column or combination of columns. In general, DATAUNLOAD and DATALOAD are best for any data manipulation against the existing database because you have the possibility to make changes.

Export/Import

Data may be needed for a test system or for another application. For example, if you are building a new application, you may want to use some of your data or tables as a sample of the actual data with which the application will need to work. Or you may want to unload data so it can be moved to another database, used by a completely different type of application, or moved to another platform. There, this data could be used for creating a report, making an analysis, or for other purposes.

Other database system

If data needs to be unloaded from or reloaded to a different platform or another type of database, the appropriate command should be used there to do the equivalent operation.

Examples

The example in Figure 12 selects specific columns from the table *"EMPLOYEE"* and unloads the data into an output file defined under the identification *"SAVE"*. Figure 13 shows the FILEDEF definition giving *"SAVE"* the file specification *"SAVE DATA A"*.

Figure 14 on page 28 and Figure 15 on page 28 show an example, which creates a new table named *"EMPLOYEE_2."* Afterwards it loads specific data rows (*"EMPNO='0002'"*) from the outfile *"SAVE"*, written by the previous DATAUNLOAD command.

```
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW;
DATAUNLOAD
   SELECT EMPNO,PHONENO,SALARY
   FROM SQLDBA.EMPLOYEE;
OUTFILE (SAVE)
```

Figure 12. SYSIN File for DBSU DATAUNLOAD (DBSUUN2 FILE)

```
/*---*/
¢FILEDEF SAVE DISK SAVE DATA A(RECFM F LRECL 80¢
¢EXEC SQLDBSU SYSIN(DBSUUN2 FILE A) SYSPRINT(DBSUUN2 LIST A)¢
```

Figure 13. EXEC File for DBSU DATAUNLOAD (DBSU1F EXEC)

```
CREATE TABLE SQLDBA.EMPLOYEE_2
      (EMPNO   CHAR(6),
       PHONENO CHAR(4),
       SALARY  DECIMAL(9,2))
  IN PUBLIC.SAMPLE;
DATALOAD TABLE SQLDBA.EMPLOYEE_2 IF POS(1-4) =¢0002¢
       EMPNO   01-06
       PHONENO 08-11
       SALARY  13-23
INFILE (SAVE);
```

Figure 14. SYSIN File for DBSU DATALOAD (DBSULOAD FILE)

```
/*--*/
¢FILEDEF SAVE DISK SAVE DATA A (RECFM F LRECL 80¢
¢EXEC SQLDBSU SYSIN(DBSULOAD FILE A) SYSPRINT(DBSULOAD LIST A)¢
```

Figure 15. EXEC File for DBSU DATALOAD (DBSULOAD EXEC)

3.2.1 Ease of Operation with Control Center

If data remains on the same platform, Control Center can help you to ease work. The Control Center function SQLTABLE uses the DATAUNLOAD function and offers a menu driven interface to provide options to:

- add or delete columns
- save the DDL needed to reload the data to the same table, a different table, or a different DB2 database
- select the DATA ONLY parameter to provide a copy of the data for any other purpose.

For a complete description of SQLTABLE refer to *Control Center for VM Installation and Operations Guide* or *Control Center for VSE Installation and Operations Guide.*

3.3 DBSU UNLOAD and RELOAD

The UNLOAD and RELOAD commands are similar to the previous DATAUNLOAD and DATALOAD commands. One difference is that they can be applied not only to tables, but also to views or dbspaces. The major difference is that the data must be (un)loaded in a system defined format. Also, UNLOAD unloads the rows sorted according to the first index created for that table. This first index is also known as the clustering index. This index is identified by a value of F or W in the column CLUSTER in the SYSTEM.SYSINDEXES table. The data is ordered by the clustering index during UNLOAD. If no indexes exist on this table, the rows are unloaded in no particular order.

In addition to containing a record for each row in the table, the UNLOAD output file also has records containing information about the table and indexes. This information is needed by the RELOAD function. UNLOAD does not unload any indexes, only the statements to recreate these indexes. No SELECT statement is used to unload the table.

Purpose

The main purpose of UNLOAD and RELOAD is reorganization for a better index. You can also choose this method if you want to change dbspace characteristics, such as PCTINDEX. However, you cannot change the characteristics of tables or columns while running this type of reorganization.

Another reason can be for balancing the I/O load between different DASD. The table or dbspace can be moved to another storage pool that owns extents on another DASD volume.

Examples

Figure 16 and Figure 17 show an example of how the UNLOAD command could be used. Figure 18 and Figure 19 show how to RELOAD a table with the PURGE parameter. PURGE identifies that the output table (table to be loaded) exists and that all existing table rows should be deleted by RELOAD TABLE processing before loading.

```
UNLOAD
TABLE (SQLDBA.ACTIVITY)
OUTFILE (ACTIVITY);
```

Figure 16. SYSIN File for DBSU UNLOAD (DBSUN3A FILE)

```
/*---*/
¢FILEDEF ACTIVITY DISK ACTIVITY DATA A4 ¢
¢EXEC SQLDBSU SYSIN(DBSUN3A FILE A) SYSPRINT(DBSUN3A LIST A)¢
```

Figure 17. EXEC File for DBSU UNLOAD (DBSUN3A EXEC)

```
RELOAD
TABLE(SQLDBA.ACTIVITY)
PURGE
INFILE (ACTIVITY)
```

Figure 18. SYSIN File for DBSU RELOAD (DBSUR4A FILE)

```
/* */
¢FILEDEF ACTIVITY DISK ACTIVITY DATA A4¢
¢EXEC SQLDBSU SYSIN(DBSUR4A FILE A) SYSPRINT(DBSUR4C LIST A)¢
```

Figure 19. EXEC File for DBSU RELOAD (DBSUR4B EXEC)

3.3.1 Ease of Operation with Control Center

Enhanced Control Center reorganization and maintenance tools support you in the different activities of a database administrator. The Control Center command SQLREORG offers a menu-driven interface and makes use of the DB2 UNLOAD and RELOAD functions. With the Control Center SQLREORG option you can choose to change the data definition values. For example, you can choose to have a different dbspace name, PCTINDEX value, or change the size of the dbspace.

For a complete description of SQLREORG refer to *Control Center for VM Installation and Operations Guide* or *Control Center for VSE Installation and Operations Guide.*

3.4 Data Restore SELECT

SELECT is a function of the Data Restore feature and allows you to select data from DB2 tables directly out of the dbextents, bypassing the database manager. With the Data Restore feature you can select tables while the database manager is online or offline.

Note: The Data Restore SELECT function does not execute local date and time user exits, nor does it execute field procedures on the selected columns. Any table without LONG columns can be specified. This restriction is true even when the database manager is online.

Data Restore SELECT writes the selected data to the DATAUNL file in a format that the DBSU DATALOAD facility can use. To help with the use of the DBSU DATALOAD facility, DBSU control statements are written to SYSPRINT.

If SELECT is used when the database manager is online, the Data Restore feature requests an exclusive lock on the dbspace. This lock ensures that no one can update data while the Data Restore feature is processing the SELECT request.

Purpose

Data Restore SELECT allows you to retrieve data while the database manager is offline, for example due to a failure. This can be of help during error recovery if no backup of the data is available.

Note: Do not rely upon this, because, for example, the ability to retrieve data from a defect disk is limited, as described in chapter 4.2.4.2, “From a Logical Error” on page 56.

Files

The Data Restore SELECT requires some input and output files:

For **VM** only:

- **SYSIN** file - contains the command to be processed. Figure 20 on page 31 shows a sample.
- **SYSPRINT** file - Data Restore feature creates a report that lists the SYSIN values, messages and results. Figure 22 on page 31 shows a sample of the SYSPRINT output.

For **VM** and **VSE**:

- **DATAUNL** - this file will contain the selected data when OUTPUT=TAPE/DASD is specified.

Examples

Figure 20, Figure 21 and Figure 22 show a sample for the SELECT function. The selected data was saved on disk (OUTPUT=DASD). In the SYSPRINT file you have additional information that you can use later for a DBSU DATALOAD.

```
CONTROL BASE=S35VMDB1 DBAPW=SQLDBAPW
SELECT  CREATOR=SQLDBA TNAME=ACTIVITY
        COLUMNS=(01,ACTKWD,03) OUTPUT=DASD
        COUNT=10
```

Figure 20. SYSIN File for Data Restore SELECT (DRFSE1A SYSIN)

```
/*---*/
¢FILEDEF DATAUNL  DISK DRFSACT DATA A(RECFM F LRECL 80¢
¢FILEDEF SYSPRINT DISK DRFSE1C SYSPRINT A¢
¢FILEDEF SYSIN DISK DRFSE1A SYSIN A¢
¢XEDIT DRFSE1A SYSIN A¢
¢XTS91001¢
```

Figure 21. EXEC File for Data Restore SELECT (DRFSE1B EXEC)

```
XTS9-143 CONTROL BASE=S35VMDB1 DBAPW=********
XTS9-143 SELECT  CREATOR=SQLDBA TNAME=ACTIVITY
XTS9-143         COLUMNS=(01,ACTKWD,03) OUTPUT=DASD
XTS9-143         COUNT=10
XTS9-143 /*
XTS9-196 Do you want to continue the SELECT    process ?
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-403 Reply is 1
XTS9-100 Data Restore feature 3.5.0
XTS9-313 SQL/DS Version 3 processed
*********************************************************************
†ACTNO†              0007-0008 FIXED
†ACTKWD†             0011-0016 CHARACTER
†ACTDESC†            0021-0040 CHARACTER
*********************************************************************
*********************************************************************
XTS9-152        10 rows selected
XTS9-007 Processing successfully completed
```

Figure 22. SYSPRINT File of Data Restore SELECT (DRFSE1C SYSPRINT)

Figure 23 on page 32 shows a sample for the DATALOAD command using the data output information from the SYSPRINT file of the previous SELECT function.

```
DATALOAD TABLE(SQLDBA.ACTIVITY)
   ACTNO    7-8   FIXED
   ACTKWD  11-16  CHAR
   ACTDESC 21-40  CHAR
INFILE(DRFSACT)
```

Figure 23. DBSU DATALOAD Command with Input from Data Restore SELECT Output

Hint: To unload data from big tables, the Data Restore SELECT function acquires some working storage. For example to SELECT data from the table *"ORDERITEM"*, you have to increase the WRKSIZE parameter on the OPTIONS statement otherwise you get the following error message, shown in Figure 24.

```
XTS9-122 Processing capacity exhausted (LMBRP043)
XTS9-152         0 rows selected
XTS9-103 Filename(         ) unknown for program(LMBRC034)
XTS9-157 Storage violation has occurred
```

Figure 24. Data Restore SELECT WRKSIZE Error Message

The value for WRKSIZE is expressed in KB and the default for SELECT is 256KB. For our example, we increased the WRKSIZE parameter as shown in Figure 25.

```
OPTIONS WRKSIZE=1000
CONTROL DBNAME=S35VMDB1
SELECT  CREATOR=SQLDBA TNAME=ORDERITEM
        COLUMNS=(01,03,OI_LNUMBER,05,07,09) OUTPUT=DASD
        COUNT=1000
```

Figure 25. SYSIN File for Data Restore SELECT with WRKSIZE Option

3.5 Data Restore UNLOAD and RELOAD

As opposed to DBSU UNLOAD and RELOAD, Data Restore UNLOAD and RELOAD do not only perform differently, but also serve different purposes.

While DBSU UNLOAD and RELOAD can both work on the same units, either dbspaces or tables, Data Restore UNLOAD can only be made from dbspaces, and Data Restore RELOAD can only load one table in one command from that unload file.

While DBSU UNLOAD and RELOAD can be used for reorganization, and UNLOAD sorts the rows according to the clustering index, Data Restore UNLOAD does no sorting at all, nor does Data Restore RELOAD.
The reason is the following: While DBSU UNLOAD and RELOAD are designed for restructure, Data Restore UNLOAD and RELOAD serve the purpose of table recovery. For restructure, performance of UNLOAD and RELOAD are equally important, but for recovery, UNLOAD will be performed much more often than RELOAD.

Therefore, the Data Restore feature improves the performance of the UNLOAD function: Data Restore UNLOAD actually unloads all the active pages of the dbspace, whereas DBSU UNLOAD unloads row after row from the table. This makes Data Restore UNLOAD faster for large tables and filled dbspaces. However, Data Restore RELOAD processing is slower because it must read the output file of Data Restore UNLOAD, which contains all data pages, and must find the rows in this file. DBSU RELOAD does not have to do this, since DBSU UNLOAD already has unloaded the rows in sequence from the table.

The difference between the Data Restore feature and DBSU is a question of which process, either unload or reload, has to analyze the pages to locate the data rows: For DBSU, the rows are found in the data pages and sorted during UNLOAD; for the Data Restore feature, the rows are found in the pages during RELOAD, and keep their sequence.

3.5.1 Data Restore UNLOAD

The UNLOAD function of the Data Restore feature enables you to perform selective backups of data. Data Restore UNLOAD allows you to unload data from DB2 dbspaces to a file with a system-defined format. With the Data Restore feature you can unload single or multiple dbspaces while the database manager is online or offline. The Data Restore UNLOAD and RELOAD file format is not compatible with DBSU UNLOAD and RELOAD.

With the UNLOAD function you can not only specify one dbspace to be unloaded, but you can also choose:

- either a list of dbspaces to be unloaded
- or to unload all dbspaces, except for a list of dbspaces

The parameters COND=INCLUDE or EXCLUDE on the UNLOAD command identify whether the list of dbspaces is to be included or excluded (INCLUDE is the default). The restriction is that only one UNLOAD statement can be specified within one SYSIN file. For the RELOAD it is possible to specify more than one statement. Figure 37 on page 41, Figure 38 on page 42 and Figure 69 on page 94 show samples for the RELOAD function with RECOVERY=YES and two RELOAD statements specified.

Purpose

With the UNLOAD function you can back up dbspaces containing critical tables more frequently than the rest of your database. By defining one table per dbspace, you can even make table-level backups.

Files

The Data Restore UNLOAD requires some input and output files:

For **VM** only:

- **SYSIN** file - contains the command to be processed and the parameters that specify what and how it will be executed. Figure 26 on page 35 shows a sample.
- **SYSPRINT** file - a report that lists the SYSIN values, messages and results. Figure 28 on page 35 shows a sample of the SYSPRINT output.

For **VM** and **VSE**:

- **ARCHIV** - this ddname, together with OPTIONS DEVICE=DASD/TAPE in the SYSIN file, identifies the file that contains the output data. Figure 27 on page 35 shows the FILEDEF definitions of the UNLOAD function.

For a complete and detailed description of this command, refer to the *Data Restore Guide*.

Example

Figure 26, Figure 27 and Figure 28 show how to UNLOAD a dbspace while the database manager is running (MODE=ONLINE). When you do this, make sure that there are no concurrent updates on the dbspace, because Data Restore UNLOAD will issue a LOCK DBSPACE command. To be sure that all updates to the dbspace are on disk, an online UNLOAD forces a checkpoint by issuing an ACQUIRE DBSPACE and a DROP DBSPACE command.

At least one unacquired dbspace should be available to complete this process.

```
OPTIONS DEVICE=DASD
CONTROL BASE=S35VMDB1 DBAPW=SQLDBAPW
UNLOAD  DBSPACE(SAMPLE) MODE=ONLINE
```

Figure 26. SYSIN File for Data Restore UNLOAD (DRFUNLOA SYSIN)

```
/*---*/
¢FILEDEF ARCHIV   DISK DRFSAMPL DATA A (RECFM VB BLOCK 32760¢
¢FILEDEF SYSIN    DISK DRFUNLOA SYSIN A¢
¢FILEDEF SYSPRINT DISK DRFUNLOA SYSPRINT A¢
```

Figure 27. EXEC File for Data Restore UNLOAD (DRFUNLOA EXEC)

```
XTS9-143 CONTROL BASE=S35VMDB1,DBAPW=********
XTS9-143 UNLOAD  MODE=ONLINE,DBSPACE=(SAMPLE),COND=INCLUDE
XTS9-143 /*
XTS9-196 Do you want to continue the UNLOAD    process ?
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-403 Reply is 1
XTS9-100 Data Restore feature 3.5.0
XTS9-309 Processing SQL/DS version 3
XTS9-160 External labeling of this unload is
XTS9-142 Base S35VMDB1 Date 04/06/96 Time 11:13:16
XTS9-013 Table SQLDBA  .ACTIVITY          may be reloaded
XTS9-013 Table SQLDBA  .DEPARTMENT        may be reloaded
........
........
XTS9-013 Table SQLDBA  .PROJECT           may be reloaded
XTS9-006 Processing DDSK1
XTS9-005        94 blocks saved
XTS9-007 Processing successfully completed
```

Figure 28. SYSPRINT File of Data Restore UNLOAD (DRFUNLOA SYSPRINT)

3.5.2 Data Restore TRANSLATE

The Data Restore command TRANSLATE can be used to increase the performance of the Data Restore RELOAD from a DB2 archive. TRANSLATE reads the DB2 archive tapes twice and creates tapes or disk files that can be used as input for the Data Restore RELOAD, as if it were a Data Restore archive

made via Data Restore BACKUP. TRANSLATE improves the performance of the Data Restore RELOAD, but is not required.

For more details, see Chapter 2.7, “Data Restore TRANSLATE” on page 14.

3.5.3 Data Restore DESCRIBE

The Data Restore DESCRIBE function reads one of the following:

- A Data Restore archive
- A Data Restore translated DB2 archive
- A Data Restore UNLOAD file

and produces a report with information about the contents of that file. From the report you can get information about the following values: date, time and names of reloadable tables.

Files

The Data Restore DESCRIBE requires some input and output files:

For **VM** only:

- **SYSIN** file - contains the command to be processed; in this case the DESCRIBE function. Figure 29 shows a sample of how this command could be used.
- **SYSPRINT** file - Data Restore feature creates a report of the contents from the Data Restore archive or Data Restore UNLOADed file. Figure 31 on page 37 shows a sample of the SYSPRINT output.

For **VM** and **VSE**:

- **ARCHIV** - one of the following:
  - Data Restore archive file to be listed
  - Data Restore UNLOAD file to be listed

Example

```
CONTROL BASE=S35VMDB1
DESCRIBE
```

Figure 29. SYSIN File for Data Restore DESCRIBE (DESCRIBE SYSIN)

Figure 30 shows a sample EXEC for the DESCRIBE function. The FILEDEF ARCHIV statement points to the file to be listed, and in this case it is an unloaded dbspace on disk.

```
/**/
¢FILEDEF ARCHIV   DISK DRFSAMPL DATA A (RECFM VB BLOCK 32760¢
¢FILEDEF SYSIN    DISK DESCRIBE SYSIN A¢
¢FILEDEF SYSPRINT DISK DESCRIBE SYSPRINT A¢
¢XTS91001¢
Exit rc
```

Figure 30. EXEC File for Data Restore DESCRIBE (DESCRIBE EXEC)

Figure 31 shows a SYSPRINT report from a DESCRIBE function of an unloaded dbspace. Message XTS9-174 shows that the dbspace was unloaded by Data Restore on June 10, 1996 at 16:49:26. The other information lists the tables in the dbspace and indicates that the DESCRIBE function was successfully completed.

```
XTS9-143 CONTROL BASE=S35VMDB1
XTS9-143 DESCRIBE
XTS9-143 /*
XTS9-196 Do you want to continue the DESCRIBE  process ?
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-403 Reply is 1
XTS9-100 Data Restore feature 3.5.0
XTS9-174 Processing S35VMDB1 unloaded on (10/06/96-16:49:26)
XTS9-013 Table SQLDBA  .ACTIVITY           may be reloaded
XTS9-013 Table SQLDBA  .DEPARTMENT         may be reloaded
XTS9-013 Table SQLDBA  .DRF_ACTIVITY       may be reloaded
XTS9-013 Table SQLDBA  .DRF_SQL_ACTIVITY   may be reloaded
XTS9-013 Table SQLDBA  .EMP_ACT            may be reloaded
XTS9-013 Table SQLDBA  .EMPLOYEE           may be reloaded
XTS9-013 Table SQLDBA  .PROJ_ACT           may be reloaded
XTS9-013 Table SQLDBA  .PROJECT            may be reloaded
XTS9-007 Processing successfully completed
```

Figure 31. SYSPRINT File of Data Restore DESCRIBE (DESCRIBE SYSPRINT)

3.5.4 Data Restore RELOAD

The Data Restore RELOAD function loads single tables from one of the following:

- a DB2 archive
- a translated DB2 archive
- a Data Restore archive
- a Data Restore UNLOAD file

The DB2 database manager must be online during Data Restore RELOAD, and applications using the database manager cannot access any table being reloaded during the reload process. For the RELOAD from an archive, the option RECOVERY=YES can be specified. This allows you to later perform a forward log recovery. Forward log recovery is not available for an UNLOAD file.

If RECOVERY=YES is specified, the log records containing the relevant DB2 statements are extracted from the database manager log (and, if LOGMODE=L, from the analyzed log archive) and written to disk.

You can RELOAD tables from a DB2 archive in two ways:

- Use the TRANSLATE function to convert the DB2 archive into the Data Restore archive format.

  You can run the TRANSLATE function at any time after you have taken the DB2 archive with the database either online or offline. Running TRANSLATE in advance will minimize the processing time when a RELOAD is required. However, you may have to TRANSLATE many DB2 archives that will never be reloaded, and you must keep the work files SYS0001, HEADER and

DIRWORK, produced during the Data Restore TRANSLATE, for a later Data Restore RELOAD. In VSE, a retention period of 0 days is **not** suitable.

- Run the RELOAD directly from the DB2 archive tape.

  Specify the options statement ARCHTYPE=SQLDS to identify the type of input tapes. The Data Restore RELOAD processing of a DB2 archive reads the archive tape twice and produces the SYS0001, HEADER, and DIRWORK work files. If you are using the Data Restore RELOAD command with a DB2 archive in a **VSE** environment, make sure you define the work files with a retention period of 0 days. This avoids the *4228I* and *4233I* messages indicating that the file already exists when running Data Restore RELOAD again.

You cannot reload the System Catalog tables.

Files

The Data Restore RELOAD requires some input and output files:

For **VM** only:

- **SYSIN** file - contains the command to be processed. Figure 32 on page 39 shows a sample.
- **SYSPRINT** file - Data Restore feature creates a report that lists the SYSIN values, messages and results. Figure 34 on page 39 shows a sample of the SYSPRINT output.

For **VM** and **VSE**:

- **ARCHIV** - one of the following:
  - input file from a Data Restore archive created by BACKUP
  - input file from a Data Restore archive produced by a Data Restore TRANSLATE function against a DB2 archive
  - workfile during RELOAD if the input is an DB2 archive
  - input file from an output of a Data Restore UNLOAD
- **ARIARCH** - input file from a DB2 archive
- **LMBRWRK** - workfile, which contains one of the following:
  - pages for tables containing LONG columns,
  - the unloaded dbspace from where the table is to be reloaded with RECOVERY=YES
- **LMBRLG1** - output, used to extract all of the changes referenced in the log files. This is only needed when the option RECOVERY=YES is specified.
- **LMBRLG2** - output, stores information about rollback LUW. This is only needed when the option RECOVERY=YES is specified.
- **LMBRLG3** - output, contains columns in LONGVARCHAR format for the updated rows. This is only needed when the option RECOVERY=YES is specified.

LMBRWRK, LMBRLG1, LMBRLG2 and LMBRLG3 will be used by the LISTLOG and APPLYLOG functions. Figure 38 on page 42 shows an example.

- **LARCHIV** - input, the log archive files or tapes
- **SYS0001** - workfile, stores the active pages of this dbspace
- **HEADER** - workfile, stores the header pages for this dbspace

- **DIRWORK** - workfile, stores the directory pages

The specifications for the work files needed with RECOVERY=YES can be found in Figure 66 on page 92.

For a complete and detailed description of this command and the RELOAD dataflow refer to *Data Restore Guide: Restoring Logical Elements of a Database*.

Example

Figure 32, Figure 33 and Figure 34 show an example using the RELOAD function from an existing table into a new table (FUNCTION=NEW) from a Data Restore UNLOAD on disk.

```
CONTROL BASE=S35VMDB1 DBAPW=SQLDBAPW
RELOAD  CREATOR=SQLDBA TNAME=ACTIVITY
        NEWTNAME=NEW_ACTIVITY
        DBSPACE=SAMPLE FUNCTION=NEW
```

Figure 32. SYSIN File for Data Restore RELOAD (DRFRE1A SYSIN)

```
/*---*/
¢FILEDEF LMBRWRK  DISK LMBRWRK  DATA A (RECFM FB BLOCK 28672 LRECL 4096¢
¢FILEDEF ARCHIV   DISK DRFSAMPL DATA A (RECFM VB BLOCK 32760¢
¢FILEDEF SYSIN    DISK DRFRE1A  SYSIN A¢
¢FILEDEF SYSPRINT DISK DRFRE1C  SYSPRINT A¢
¢XEDIT DRFRE1A SYSIN A¢
¢XTS91001¢
```

Figure 33. EXEC File for Data Restore RELOAD (DRFRE1B EXEC)

```
XTS9-143 OPTIONS RECOVERY=NO,DEVICE=DASD
XTS9-143 CONTROL BASE=S35VMDB1
XTS9-143 RELOAD  CREATOR=SQLDBA,TNAME=ACTIVITY,FUNCT=NEW
XTS9-143  DBSPACE=SAMPLE
XTS9-143  NEWTNAME=NEW_ACTIVITY
XTS9-143 /*
XTS9-196 Do you want to continue the RELOAD    process ?
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-403 Reply is 1
XTS9-100 Data Restore feature 3.5.0
XTS9-174 Processing S35VMDB1 unloaded on (04/06/96-13:48:47)
XTS9-102       106 rows loaded, procedure completed
XTS9-128       106 rows loaded into (SQLDBA.NEW_ACTIVITY)
XTS9-101         1 tables successfully processed
XTS9-314 COMMIT WORK successful for reload of data, creating
         required objects
CONNECT SQLDBA  ;
COMMIT WORK ;
XTS9-007 Processing successfully completed
```

Figure 34. SYSPRINT File of Data Restore RELOAD (DRFRE1C SYSPRINT)

Guest Sharing

RELOAD is one of the very few Data Restore functions that can run in a VM/VSE Guest Sharing environment. It can only run if the OPTIONS statement RECOVERY=NO is specified. Figure 70 on page 94 shows a sample JCL that was used to reload *"SQLDBA.ACTIVITY"* into the VM database *"S35VMDB1"* from a Data Restore feature archive of the VSE database *"SQLVSE02"*.

The CONTROL statement DBNAME=SQLVSE02 defines the name of the database from which this backup or unload originated. The PARM='DBNAME(S35VMDB1)' parameter, defines the target database for the reload of this table.

3.6 Data Restore RELOAD with Forward Recovery

Data Restore RELOAD is the only way to apply log recovery on a table level. When restoring a complete database you can apply log recovery, but in some cases restoring a complete database might be an unreasonably large effort. With Data Restore RELOAD, you can even specify to apply log recovery up to a certain point in time.

You can also use Control Center to execute a RELOAD with forward recovery. For a complete example of Control Center using Data Restore RELOAD function, please refer to C.8, "Example of Data Restore RELOAD Using Control Center" on page 104.

Here we are going to show you a sample scenario how a table can be reloaded from a complete Data Restore BACKUP and also apply the log archives and the current log up to a certain point in time.

The sequence of operations was:

1. COLDLOG to clear old log information
2. Data Restore BACKUP or DB2 archive of complete database
3. Modifications against the table *"SQLDBA.PROJECT"*
4. RELOAD the affected table from Data Restore archive or DB2 archive
5. LISTLOG to display all modifications
6. APPLYLOG to apply changes from log (to point in time if "END=" parameter is specified).

For the original data before any modification see Figure 58 on page 88.
The modifications are:

```
UPDATE SQLDBA.PROJECT SET PROJNAME=¢SYSTEMS SUPPORT UPD1¢
   WHERE PROJNO=¢OP2010¢
UPDATE SQLDBA.PROJECT SET PROJNAME=¢OPERATION UPD2¢
   WHERE PROJNO=¢OP1010¢
UPDATE SQLDBA.PROJECT SET PROJNAME=¢USER EDUCATION UPD3¢
   WHERE PROJNO=¢IF2000¢
DELETE FROM SQLDBA.PROJECT WHERE PROJNO = ¢MA2100¢
```

Figure 35. Modifications against SQLDBA.PROJECT

The FILEDEF statement for Data Restore RELOAD from archive (and log archive, if used) should be adjusted to the FILEDEF used when the archive (and log archive) was created. Make sure you adjust the options for record length, and blocking factor. The device name TAP1 should be used in VM for the log archives and TAP2 for the archive. Special care should be taken on the FILEDEF for the log archive tapes, as these tapes are only requested after the archive tapes and if any error occurs, the complete tape set has to be read again, losing precious time. For tapes from a DB2 archive, these tapes have to be read twice. For the exec used in VM to reload a table refer to Figure 59 on page 88, for the SYSPRINT see Figure 60 on page 89. For the JCL used in VSE to reload a table refer to Figure 67 on page 92 and Figure 66 on page 92. If a DB2 archive is used as input, specify SYS007 as the input assignment. If the ARCHIV workfile is on disk, you must also specify OPTIONS DEVICE=DASD. For this JCL refer to Figure 68 on page 93.

RELOAD reads the archive tapes and restores the table as it was at the point in time the archive was taken. Additionally it writes the log entries that are relevant for this table into workfiles. Therefore you are prompted to mount the log archive tapes (if any) and have the choice to skip them as shown in Figure 36. For the complete output of this function refer to Figure 60 on page 89. Those workfiles are later used to apply forward recovery on this table using the Data Restore feature functions LISTLOG and APPLYLOG described below. For a comparison between filtered log recovery and APPLYLOG with forward log recovery to a certain point in time, see chapter 2.5, “Log Recovery” on page 13.

```
XTS9-202 Processing SQL/DS archive (timestamp=05/28/96 11:29:22)
XTS9-182 Following files are needed for recovery
XTS9-195 ARCHIVE      currently mounted
XTS9-180 LARCHIVE     at 05/28/96 11:52:37
XTS9-179 Current log
 .......
XTS9-183 Please mount larchive at 05/28/96 11:52:37
XTS9-407 Enter 0(CANCEL),1(CONTINUE) or 111(SKIPFILE)
 .......
XTS9-184 Processing current log
XTS9-407 Enter 0(CANCEL),1(CONTINUE) or 111(SKIPFILE)
```

Figure 36. Forward Recovery Prompt during Data Restore RELOAD

Figure 37 and Figure 38 on page 42 show how to reload two tables with RECOVERY=YES specified in VM; for VSE, see Figure 69 on page 94.

```
OPTIONS RECOVERY=YES CONFIRM=NO
CONTROL BASE=S35VMDB1
RELOAD  CREATOR=SQLDBA,TNAME=EMP_ACT,FUNCT=REPLACE
RELOAD  CREATOR=SQLDBA,TNAME=PROJ_ACT,FUNCT=REPLACE
```

Figure 37. SYSIN File for Data Restore RELOAD Tables and RECOVERY=YES

```
/*-----------------------------------------------------------------*/
/*- Reload procedure with forward Recovery                        -*/
/*- input drf backup tape                                         -*/
/*-----------------------------------------------------------------*/
¢FILEDEF ARCHIV   DISK ARCHIV DATA T (RECFM VB BLOCK 32760¢
¢FILEDEF LARCHIV  TAP1 SL            (RECFM FB BLOCK 28672 LRECL 4096¢

¢FILEDEF LMBRLG1 DISK LMBRLG1 DATA T (RECFM VB BLOCK 32760¢
¢FILEDEF LMBRLG2 DISK LMBRLG2 DATA T (RECFM VB BLOCK 32760¢
¢FILEDEF LMBRLG3 DISK LMBRLG3 DATA T (RECFM VB BLOCK 32760¢

¢FILEDEF LMBRWRK DISK LMBRWRK DATA T (RECFM FB BLOCK 28672 LRECL 4096¢

¢FILEDEF SYSIN   DISK  DRFRELD   SYSIN  A¢
¢FILEDEF SYSPRINT DISK DRFRELD   SYSPRINT A¢
¢XTS91001¢
Exit rc
```

Figure 38. EXEC File for Data Restore RELOAD Tables and RECOVERY=YES

```
XTS9-143 OPTIONS RECOVERY=YES CONFIRM=NO
........
........
XTS9-102      1242 rows loaded, procedure completed
XTS9-128        74 rows loaded into (SQLDBA.EMP_ACT)
XTS9-128        77 rows loaded into (SQLDBA.PROJ_ACT)
XTS9-128       572 rows loaded into (DATARFTR.SYSCOLUMNS)
XTS9-128        69 rows loaded into (DATARFTR.SYSCATALOG)
XTS9-128       190 rows loaded into (DATARFTR.SYSTABAUTH)
XTS9-128        90 rows loaded into (DATARFTR.SYSINDEXES)
XTS9-128         7 rows loaded into (DATARFTR.SYSVIEWS)
XTS9-128        15 rows loaded into (DATARFTR.SYSKEYCOLS)
XTS9-128        11 rows loaded into (DATARFTR.SYSKEYS)
XTS9-128       115 rows loaded into (DATARFTR.SYSUSAGE)
XTS9-128        22 rows loaded into (DATARFTR.SYSCOLAUTH)
XTS9-101        11 tables successfully processed
XTS9-314 COMMIT WORK successful for reload of data, creating
         required objects
CREATE INDEX †SQLDBA†.†PROJNOIN† ON †SQLDBA†.†EMP_ACT† (†PRO
JNO† ASC ) PCTFREE=10;
COMMIT WORK ;
CREATE INDEX †SQLDBA†.†EMPNOIN† ON †SQLDBA†.†EMP_ACT† (†EMPN
O† ASC ) PCTFREE=10;
COMMIT WORK ;
ALTER TABLE †SQLDBA†.†PROJ_ACT† ADD PRIMARY KEY (†PROJNO† AS
C ,†ACTNO† ASC ,†ACSTDATE† ASC ) PCTFREE=10;
COMMIT WORK ;
ALTER TABLE †SQLDBA†.†EMP_ACT† ADD FOREIGN KEY †R_EMPLY3† (†
EMPNO†) REFERENCES †SQLDBA†.†EMPLOYEE† ON DELETE CASCADE ;
COMMIT WORK ;
ALTER TABLE †SQLDBA†.†EMP_ACT† ADD FOREIGN KEY †R_PROACT† (†
PROJNO†,†ACTNO†,†EMSTDATE†) REFERENCES †SQLDBA†.†PROJ_ACT† O
N DELETE RESTRICT;
COMMIT WORK ;
ALTER TABLE †SQLDBA†.†PROJ_ACT† ADD FOREIGN KEY †R_PROJ2† (†
PROJNO†) REFERENCES †SQLDBA†.†PROJECT† ON DELETE RESTRICT;
COMMIT WORK ;
CONNECT SQLDBA  ;
GRANT SELECT ON †SQLDBA†.†EMP_ACT† TO †PUBLIC†;
COMMIT WORK ;
GRANT SELECT ON †SQLDBA†.†PROJ_ACT† TO †PUBLIC†;
COMMIT WORK ;
CONNECT SQLDBA  ;
COMMIT WORK ;
XTS9-183 Please mount larchive at 06/27/96 14:56:15
........
XTS9-007 Processing successfully completed
```

Figure 39. SYSPRINT File for Data Restore RELOAD Tables and RECOVERY=YES

3.6.1 Data Restore LISTLOG

Data Restore LISTLOG lists the DB2 statements extracted from the log (and the log archive) during the RELOAD operation with forward recovery. With this list you can determine exactly what operations should not be performed and where the recovery (Data Restore APPLYLOG) should stop.

Note: The LISTLOG function stops when it recognizes a DROP TABLE, ALTER TABLE or DROP DBSPACE command. There is no way to list the changes made after that. Figure 42 on page 45 shows a sample of how this message is presented to the user.

Figure 40 on page 44 shows the JCL that was used to perform this function in VSE. The JCL of XTS9DLBL is shown in Figure 66 on page 92. Figure 64 on page 91 shows the EXEC that was used to perform this function in VM. The sample output can be found in Figure 41 on page 45 . It lists the modifications shown in Figure 35 on page 40. For the file definitions please refer to chapter 3.5.4, “Data Restore RELOAD” on page 37.

```
* $$ JOB JNM=DRFLSTLG,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFLSTLG                LISTLOG procedure
// LIBDEF *,SEARCH=(PRD2.SQL350,PRD2.RCV350)
// EXEC PROC=SQLVSE02       dlbl for database dbextents
// EXEC PROC=XTS9DLBL       dlbl for data restore feature workfiles
// EXEC XTS91001,SIZE=AUTO
CONTROL DBAPW=SQLDBAPW
LISTLOG
/*
/&
* $$ EOJ
```

Figure 40. JCL File for Data Restore LISTLOG

```
XTS9-143 CONTROL DBAPW=********
XTS9-143 LISTLOG
XTS9-143 /*
XTS9-196 Do you want to continue the LISTLOG   process ?
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-403 Reply is 1
XTS9-100 Data Restore feature 3.5.0

C 00000512 1996-151-15-32-37-908640
UPDATE †SQLDBA†.†PROJECT† SET †PROJNO† =¢OP2010¢,
†PROJNAME† =¢SYSTEMS SUPPORT UPD1¢,†DEPTNO† =¢E21¢,
†RESPEMP† =¢000100¢,
†PRSTAFF† =   4,00 ,†PRSTDATE† =1982-01-01,
†PRENDATE† =1983-02-01,†MAJPROJ† =¢OP2000¢
WHERE †PROJNO† =¢OP2010¢

C 00000513 1996-151-15-32-55-265424
UPDATE †SQLDBA†.†PROJECT† SET †PROJNO† =¢OP1010¢,
†PROJNAME† =¢OPERATION UPD2¢,†DEPTNO† =¢E11¢,
†RESPEMP† =¢000090¢,†PRSTAFF† =   5,00 ,
†PRSTDATE† =1982-01-01,
†PRENDATE† =1983-02-01,†MAJPROJ† =¢OP1000¢
WHERE †PROJNO† =¢OP1010¢

C 00000514 1996-151-15-33-16-747088
UPDATE †SQLDBA†.†PROJECT† SET †PROJNO† =¢IF2000¢,
†PROJNAME† =¢USER EDUCATION UPD3¢,†DEPTNO† =¢C01¢,
†RESPEMP† =¢000030¢,†
†PRSTAFF† =   1,00 ,†PRSTDATE† =1982-01-01,
†PRENDATE† =1983-02-01,†MAJPROJ† =NULL
WHERE †PROJNO† =¢IF2000¢

C 00000516 1996-151-15-34-07-382272
DELETE FROM †SQLDBA†.†PROJECT† WHERE †PROJNO† =¢MA2100¢
XTS9-007 Processing successfully completed
```

Figure 41. SYSPRINT File of Data Restore LISTLOG

```
XTS9-196 Do you want to continue the LISTLOG   process ?
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
1
XTS9-100 Data Restore feature 3.5.0
XTS9-185 Forward recovery stopped due to DROP TBL
XTS9-186 Timestamp of statement is 1996-157-09-06-57-704112
XTS9-007 Processing successfully completed
```

Figure 42. Data Restore LISTLOG Termination

3.6.2 Data Restore APPLYLOG

Data Restore APPLYLOG applies the DB2 statements that were extracted from the log (and the log archive) during the RELOAD operation. The forward recovery can be applied completely or stopped at a certain point in time. When log recovery stops it cannot be resumed. It stops automatically when it has:

- Reached a certain predefined point in time
- Encountered a DROP TABLE, ALTER TABLE or DROP DBSPACE command

For the difference between filtered log recovery and APPLYLOG with forward log recovery to a certain point in time, see chapter 2.5, “Log Recovery” on page 13.

The command syntax is identical to the one that was used for the LISTLOG. Only the options in VSE or SYSIN for VM have to be changed. For the file definitions please refer to chapter 3.5.4, “Data Restore RELOAD” on page 37. Figure 43 shows a sample of the options in VSE or the SYSIN in VM, used to specify until what point in time these records should be applied.

```
CONTROL DBAPW=SQLDBAPW
APPLYLOG END=1996-151-15-33-16-747088
```

Figure 43. SYSIN File for Data Restore APPLYLOG

The “END=” specifies the end of the apply procedure, and the record with this corresponding timestamp is not applied. Refer to Figure 65 on page 91 for the output of the rows that were affected, in SQLDBA.PROJECT by the previous APPLYLOG. As you can see, the last two DB2 commands (UPDATE and DELETE) from Figure 35 on page 40 and Figure 41 on page 45 have not been applied.

Note: It is **not** possible to resume the log application once it has ended through

- either the “END=” statement
- or the detection of a DROP TABLE, ALTER TABLE or DROP DBSPACE command

whichever comes first.

3.7 Summary

3.7.1 Compatibilities

DBSU DATAUNLOAD/DATALOAD, DBSU UNLOAD/RELOAD, and Data Restore UNLOAD/RELOAD are pairs of functions that are in themselves compatible. You cannot use the output of one pair as input for another.

The Data Restore SELECT function is compatible with the DBSU DATALOAD function. You can SELECT data via Data Restore and load the data with the DBSU DATALOAD. The Control Center SQLTABLE function is compatible with, and makes use of, the DBSU DATAUNLOAD function.

The Control Center SQLREORG function is compatible with, and makes use of, the DBSU UNLOAD and RELOAD functions.

The Data Restore UNLOAD and RELOAD functions are incompatible with the DBSU UNLOAD and RELOAD functions, as described in chapter 3.5, “Data Restore UNLOAD and RELOAD” on page 32.

Table 4 on page 47 gives you an overview of the compatibilities among the methods to unload or reload data on a DB2 database.

Table 4. Compatibilities Data Unload and Load					
	... can be **Input** to:				
Output from ...	DBSU DATALOAD	DBSU RELOAD	Data Restore DESCRIBE	Data Restore RELOAD	Control Center SQLREORG
DBSU DATAUNLOAD	**X**	-	-	-	-
DBSU UNLOAD	-	**X**	-	-	**X**
Data Restore UNLOAD	-	-	**X**	**X**	-
Data Restore SELECT	**X**	-	-	-	-
Control Center SQLTABLE	**X**	-	-	-	-
Control Center SQLREORG	-	**X**	-	-	**X**
DB2 archive	-	-	-	**X**	-
Translated DB2 archive	-	-	**X**	**X**	-
Data Restore BACKUP	-	-	**X**	**X**	-

3.7.2 Possibilities

Table 5 on page 48 gives you an overview of the methods and their possibilities offered to unload or reload data on a DB2 database.

Reorganization can be achieved also with other means; we indicate here just the main purpose.

For “Data Restore RELOAD from TRANSLATE” (translated DB2 archive), see column “RELOAD from BACKUP.”

Table 5. Possibilities Data Unload and Load

	DBSU				Data Restore					Control Center	
Function	DATAUNLOAD	DATALOAD	UNLOAD	RELOAD	SELECT	UNLOAD	RELOAD from BACKUP	RELOAD from DB2 archive	RELOAD from UNLOAD	SQLTABLE	SQLREORG
Unit											
row	X	X	-	-	X	-	-	-	-	X	-
table	-	-	X	X	-	-	X	X	X	X	X
dbspace	-	-	X	X	-	X	-	-	-	-	X
DML Data Manipulation Language											
user defined	X	X	-	-	-	-	-	-	-	X	-
DDL Data Definition Language											
user defined columns	X	X	-	-	-	-	-	-	-	X	-
user defined table	-	-	-	X	-	-	X	X	X	-	-
user defined dbspace	-	-	-	X	-	-	X	X	X	-	X
Reorganization											
free empty pages	-	-	-	-	-	-	-	-	-	-	X
to other dbspace	-	X	-	X	-	-	X	X	X	-	X
user defined (order by)	X	-	-	-	-	-	-	-	-	-	-
system defined (clustering index)	-	-	X	-	-	-	-	-	-	X	X
Export / Import data											
DB2	X	X	X	X	X	X	X	X	X	X	X
non- DB2	X	X	-	-	-	-	-	-	-	X	-
Database status											
online	X	X	X	X	X	X	X	X	X	X	X
offline	-	-	-	-	X	X	-	-	-	-	-
Recover log											
apply all	-	-	-	-	-	-	X	X	-	-	-
to point in time	-	-	-	-	-	-	X	X	-	-	-

Chapter 4. How to Recover from Failures

This chapter describes some errors that can be encountered and how to recover from these situations.

4.1 Failures to be Considered

Different types of failures can occur in a relational database management system:

- Power or Processor

  Because of a power or processor error, the DB2 database manager can end abnormally.

- Disk

  A DASD can give hardware errors, and there might be cases where this DASD should be replaced. The DB2 database manager can end abnormally.

- Software

  Not only the operating system can end abnormally, but there may be some conditions where the database manager decides to end abnormally.

- Application or User Logic

  An application can fail and lead to a situation where some tables have been updated and committed and others that should have been are not. Interactive users can make changes and by error request an update or delete that was not intended.

The various failure sources can lead to any of the following problems, which can happen alone or in combinations:

- An LUW completion or cancellation requested by the administrator
- The database manager does not come up again
- The database manager ends abnormally
- Data on disk is damaged or unreadable, partially or completely
- Data in tables is in error, and the problem is not detected for quite some time, that is, for several archives

4.1.1 Power or Hardware Failure

From a failure where the complete system comes down, the DB2 database manager normally recovers after the UNDO/REDO process. Only the LUWs that were currently active when the failure occurred would have to be restarted. In some cases this is not so obvious and needs some investigation to be sure that the data in all tables is consistent with what is expected.

4.1.2 DASD Failure

Here you need to determine what type of disk or extent is affected.

4.1.2.1 Log Disk Failure

If single logging is used, any I/O error causes the database manager to end. With dual logging, the database updates are recorded in two log extents. The database manager continues running as long as it can read and write from one of the log disks. It is recommended to allocate these log disks on different DASD, as an unrecoverable error is unlikely to occur on both DASD at the same time.

In **VM**, you should additionally make sure that the database′s 191 minidisk is on another DASD, because this also contains a copy of the log history file, and this A-disk file is automatically used during a restore, if the log history area is unusable due to a log failure.

Note: There might be a situation where the database should be restored on a different system, or that all dbextents should be replaced. Refer to chapter 4.2.1, “Recover from System Failure” on page 52 for a detailed explanation of saving the log history information.

To recover from a damaged log disk, follow the procedure as described in *DB2 System Administration: Log Reconfiguration*.

4.1.2.2 Directory Disk Failure

To reduce the probability of failures of the directory disk (BDISK), always end the database manager with the SQLEND parameter DVERIFY, to have a directory verification. This function checks for inconsistencies in the directory, and thus allows you to avoid archiving an inconsistent directory.

Note: It is especially recommended to perform an SQLEND DVERIFY just for that purpose after having loaded a large quantity of data or after critical data has been updated. This verification can shorten the period between an inconsistency and its discovery.

Physical Error

The only way to recover from a physical error on the directory disk, is to restore any archive and, dependent on the logmode, recover the log (and log archives, if LOGMODE=L).

Logical Error

There are some undocumented built-in tools in DB2 that can help to recover from logical errors on the directory disk (BDISK) such as pointing to a wrong page. Because these tools are very powerful and if misused can destroy the entire database, these commands or procedures should never be run unless explicitly requested by the IBM Support Center. Before starting any of these tools, make sure you have an archive of the complete database just in case anything goes wrong.

4.1.2.3 Data Disk Failure

Chapter 4.2.4, “Data Recovery” on page 55 gives you an overview of procedures that can be used to recover as much data as possible, should a data disk be damaged.

Physical Error

The sequence of operations to replace a data disk is described in *DB2 System Administration: Replacing a Dbextent* for VSE or for VM in *DB2 System Administration: Replacing a Database Minidisk*.

Logical Error: Data or Index Page Corruption

A data or index page corruption can only be detected when this page is actually requested by the database manager. When a corruption is detected, it may be possible that the restore of your database or storage pool will not help, because this corruption was already present when the backup was taken.

Note: Therefore it is recommended to keep different sets of backups and the logs in between, so that it may be possible to restore your database to the point where it was free of corruption, and then apply forward recovery with the logs if they are available.

If only the **index** has been corrupted, a drop and create index might correct the problem.

The procedure for **data pages** is more complicated. Technical help from the IBM Support Center using special programs might correct such a problem, depending on the kind of corruption, but such programs should only be executed in cooperation with someone at the IBM Support Center.

4.1.3 Operating System Abend

From an operating system abnormal end, such as from a power or hardware failure, the DB2 database manager normally recovers after the UNDO/REDO process. Only the LUWs that were currently active when the failure occurred would have to be restarted. In some cases this is not so obvious and needs some investigation to be sure that the data in all tables is consistent with what is expected.

4.1.4 DB2 Failure

Due to a problem with the DB2 software, in an extreme situation a corruption could occur in the database. If the database becomes corrupted, the problem may not be very obvious and may not be noticed until several archives have been taken.

4.1.5 Application or User Logic Error

A problem may occur with an application and lead to a situation where some tables have been updated and committed and others that should have been are not. This situation requires investigation to restore the data to a consistent state, as it was before the application was started or to a point that all data is synchronized. A similar situation can happen when interactive users make changes and by error request an update or delete of data that was not intended.

Note: Care should be taken with the use of non-recoverable storage pools, as the database manager does not undo INSERT, UPDATE, PUT, DELETE statements that were successfully executed. For a detailed description of characteristics of non-recoverable storage pools see *DB2 System Administration: Special Topics in Recovery Design*.

4.2 Procedures to Recover from Failures

4.2.1 Recover from System Failure

There might be an occasion where the complete operation has to be moved to another system. In such a case, you would not only need a backup of the database, but also of all other disks that are related to the operating system and the database. In most cases you would restore an image copy of all DASDs.

Recommendation: Before making copies of all DASDs, shut down the database manager so that all data is written to disk.

If you need to move **only the database** to another system, you will need some files or jobs to restart your database. These files and jobs are related to the database but are neither saved by the DB2 ARCHIVE nor by the Data Restore BACKUP process. These are the database definitions and the log history file.

Database Definitions

In **VM** these files are:

- *userID* DIRECT - the VM system directory of the database userID is required to know the exact size of each extent.

  Note: Restore fails if the extents are too small, and will not make use of the additional space if an extent is too large.

- *userid* SQLDBN - this file keeps track of the database name, the name of the Shared Segments set to address the common DB2 code and the addressing mode with which the database is running.

- *dbname* SQLFDEF - this file keeps the FILEDEF and CP LINK statements to access all database extents. This file is modified by the database manager whenever an add, delete or copy dbextent is requested. It is recommended to add a comment in this file for each extent, describing the minidisk size.

- *dbname* SQLPARM - this file contains the database startup parameters. These parameters may also have been included in the database userID′s PROFILE EXEC, or another EXEC called by the PROFILE EXEC.

- *dbname* ARISPOOL - if VMDSS has been implemented, this file keeps information about which options have been set for each dbextent.

- *dbname* SQLDBGEN - this file keeps information that was used for the generation of the database. It is **not** required for restore, but it is good practice to keep it updated to reflect the current status of extents, pools and dbspaces.

  Note: Especially important is to know the size of the internal dbspaces and in which pool they are, which is at the end of this file or of any job adding a dbspace, because that size cannot be queried through a SHOW POOL command.

In **VSE** these jobs are:

- ARIS35CD - this job contains all statements to define the VSAM data sets for the database extents. The name can be different depending on which version and release of DB2 (or SQL/DS) you installed first; in this example, SQL/DS V3.5 was installed first, and then migrated to DB2 V5.1. It is recommended to keep this job and maintain it when adding or deleting

extents. It is anyway good practice that this job contains a DELETE statement in front of each DEFINE, as the example in the installation manual shows, so that it can easily be re-run in case any single step creates a return code.

As an alternative you can restore a backup of the VSAM catalog containing the database extents; but this must reflect the actual status.

Note: Restore fails if the extents are too small, and will not make use of the additional space if an extent is too large.

- *dbname*.PROC in the DB2 library (PRD2.SQL350, PRD2.DB2510 or other). This PROC contains the database name and all DLBL statements identifying the dbextents. To keep track of these definitions when adding or deleting extents, punch this PROC after each change. You can also maintain your installation job; in our case this is ARIS35DB because SQL/DS V3.5 was installed first, and then migrated to DB2 V5.1.
- *ddname* - this job starts the database and contains the startup parameters currently being used on the database. Some of these parameters may have been catalogued into a library in a macro referenced by the startup job.
- ARISDBG.A in the DB2 library (PRD2.SQL350, PRD2.DB2510 or other). This macro keeps information that was used for the generation of the database. During the installation, this macro might have been updated, or it had been copied into an installation job. It is **not** required for restore, but it is good practice to keep it updated to reflect the current status of extents, pools and dbspaces.

  Note: Especially important is to know the size of the internal dbspaces and which pool they are in, which is at the end of this file or of any job adding a dbspace, because that size cannot be queried through a SHOW POOL command.

Log History File

In **VM** this file is:

- ARIHSDS ARCHIVE - this file keeps information about the log history.

This file keeps the history information, which is also available in the last page of the current log, and which can be displayed with the DB2 operator command SHOW LOGHIST.

In **VM** the log history file is located on the work disk, default 191, of the database manager. This A-disk file is automatically used during a subsequent restore, if the log history area is unusable due to a log failure.
When a COLDLOG RECONFIGURE is performed, this file is copied to ARIHSDS PRECLDLG to ensure recoverability.

In **VSE** the log history information is only available in the current log. To back up this information a VSAM backup should be made offline of the log file LOGDSK1. After disaster recovery this backup should be restored and COLDLOG REFORMAT should be performed to clear all other log information.

If you restore the database to another system, or when you have replaced all disks, you cannot apply the log archive tapes without a recent copy of the log disk or of the externalized history file.
For LOGMODE=L, this is a question of to which point in time (which log archive tape) you plan to fall back in case of a disaster. If you want to be able to

recover also the latest log archive, then whenever an archive **or log archive** is taken, you might want to immediately afterwards make a copy of the ARIHSDS ARCHIVE file.
For LOGMODE=A, on a different system you can only restore archives taken offline. Because at the begin archive checkpoint LUWs can have been active, an online archive would need forward recovery using the current log to get into a consistent state. Therefore, the log history file is less important for LOGMODE=A.

4.2.2 Recover from Directory Failure

If the directory is corrupted or the database is down and cannot be restarted due to a disk error, there are two ways to recover:

- Either with the Data Restore feature:
  1. Define a new or different extent to replace the damaged one
  2. Format it (in VSE use Data Restore FORMAT)
  3. Restore a Data Restore feature archive with the Data Restore RESTORE command. This can be either an archive taken with Data Restore BACKUP or a translated DB2 archive.
  4. Start the database with startup parameter STARTUP=U.
- Or using DB2 facilities:
  1. Define a new or different extent to replace the damaged one
  2. Format the extent and restore a DB2 archive:
     In **VM** format it and restart with the parameter STARTUP=R.
     In **VSE**,
     - either use the Data Restore FORMAT to format the new extent only and restart with STARTUP=F,
     - or just restart with STARTUP=R formatting all extents.

In all cases, when restarting use the same LOGMODE as before. If this was LOGMODE=A, the log is synchronized. If this was LOGMODE=L, you are prompted for subsequent log files to be restored, and the log is synchronized.

4.2.3 Recover from Database Corruption

Should you detect a user or program error, disable the dbspace containing the table to prevent more updates to the damaged table or to prevent users from getting incorrect data. This DISABLE DBSPACE command takes the specified dbspace offline so no access is allowed. Once you have determined how to fix this user error, you can use the ENABLE DBSPACE command to bring the dbspace back online and fix the problem.

The same procedure applies if there are persistent DB2 abnormal terminations in DBSS. In this case the database manager would terminate and display the message ARI040E which would identify a module whose first four characters are ARIY. To avoid more corruption, or until the problem is corrected, disable all the dbspaces involved.

Another way to recover from abnormal DBSS terminations would be to restore a DB2 archive, Data Restore archive or user archive and apply the log tapes with *Filtered Log Recovery.* This allows you to exclude some LUWs that affected the corrupted dbspace and might have led to the error.

The command sequence for disabling a dbspace and for forward log recovery is described in *DB2 Diagnosis and Reference: Recovering from DBSS Errors*.

4.2.4 Data Recovery

4.2.4.1 From a Physical Error

You can replace the disk and restore the database as follows:

- Either with the Data Restore feature:
  1. Define a new or different extent to replace the damaged one
  2. Format it (in VSE use Data Restore FORMAT)
  3. Restore a Data Restore feature archive with the Data Restore RESTORE command (full database or just one storage pool). This can be either an archive taken with Data Restore BACKUP or a translated DB2 archive.
  4. Start the database with startup parameter STARTUP=U.
- Or using DB2 facilities:
  1. Define a new or different extent to replace the damaged one
  2. Format the extent and restore a DB2 archive:
     In **VM** format it and restart with the parameter STARTUP=R.
     In **VSE**,
     - either use the Data Restore FORMAT to format the new extent only and restart with STARTUP=F,
     - or just restart with STARTUP=R formatting all extents.

In all cases, when restarting use the same LOGMODE as before. If this was LOGMODE=A, the log is synchronized. If this was LOGMODE=L, you are prompted for subsequent log files to be restored, and the log is synchronized.

For more information about restoring a storage pool and log recovery, please refer to 2.5, “Log Recovery” on page 13.

You can also try the following procedure, but it might not always work and can take more time than the storage pool level recovery or even the database restore:

1. Drop the dbspace where the damaged area is
2. Add a new extent to the pool to replace the damaged one
3. Delete the damaged extent. This will move the data from the damaged extent to the space available on the new extent in the same storage pool.
4. Create the dbspace again
5. Acquire the dbspace
6. Reload the dbspace that you dropped previously using the Data Restore RELOAD with forward recovery from an archive:
   - Take a DB2 archive or a Data Restore archive and list all tables of the dropped dbspace using Data Restore DESCRIBE.
   - Data Restore RELOAD all the tables from the archive with one RELOAD command for each table, all commands within the same job (JCL or SYSIN file). With RECOVERY=YES the log recovery is prepared. With LOGMODE=L and log archive on tapes, you have to mount them.

- Apply forward recovery with Data Restore APPLYLOG.

4.2.4.2 From a Logical Error

If the Dbspace can be Dropped

You can:

- Either repair the database in the following way:
  1. Drop the dbspace for which there is a logical error
  2. Create the dbspace again
  3. Acquire the dbspace
  4. Reload all tables of the dbspace that you dropped previously using the Data Restore RELOAD with forward recovery from an archive, as described in chapter 4.2.4.1, “From a Physical Error” on page 55.
- Or restore the storage pool or the database as described in chapter 4.2.4.1, “From a Physical Error” on page 55. Depending on the number of tables in the dbspace in error, this may be the quicker way.

If LOGMODE=Y you cannot use forward recovery. However, the Data Restore feature does let you recover some data from the database with the SELECT command. The UNLOAD command cannot be used because it will end abnormally when it tries to read the corrupted data. The output from such a partially unloaded dbspace is not usable for the RELOAD process. Try to unload data using the SELECT command from the tables that reside in the dbspace, that report the error. The SELECT command ends at the point where it tries to read the data in error. Neither the Data Restore feature nor DB2 can bypass the damaged area and continue unloading the required data. Some of this unloaded data might be useful to help determine the differences between the table reloaded from the archive and the output of the previously executed SELECT.

If the Dbspace cannot be Dropped

There are cases where DB2 does not allow you to drop a dbspace but abnormally ends instead. In this situation, you can:

- Either restore the database from archive tapes and consecutive log files, including the current log. Follow the description in chapter 4.2.2, “Recover from Directory Failure” on page 54.
- Or disable the dbspace and contact the IBM Support Center for help. Acquire a new dbspace and Data Restore RELOAD and APPLYLOG the individual tables of this dbspace from an archive as described before. **Warning:** The tables must be reloaded with a new name, because you cannot have duplicate table names in the catalog. This means that applications, pointing to the tables in the disabled dbspace, will need to be changed, preprocessed and compiled.

Chapter 5. Recovery Strategies

In this chapter you will find recommendations, considerations and scenarios to choose from, so that you can define your own archive and recovery strategy. A migration part gives you hints how you can realize a change in your strategy.

5.1 Recommendations

Taking backups and never having to use them, is often considered as a waste of time. But whenever you get a situation where a backup has to be restored partially or completely, you will appreciate its value.

Have a Strategy

That is why it is always recommended to have a good recovery strategy. It should not only be defined once, but also regularly reviewed and adapted to the current needs and to the experiences made when recovery has been required.

This recovery strategy should cover both backups and the possible restore scenarios. It should comprise recovery on the same system, but also on another system. Even a disaster backup and recovery should be considered, where the whole environment can be recreated from scratch on another system, for example in an IBM Backup and Recovery Center.

Test Backup And Restore

The procedures and the backups should also be tested, for example a recovery should be tried from the backups actually taken either on the same system, or if possible on a different system. If these tests are performed, they might reveal a need for changes in your archiving and recovery procedures, so that you will be able to benefit from these in case you need them.

Keep Several Backups

Keeping different sets of backups, and if applicable, log archives, may give you the possibility to return to an earlier point in time than the latest archive. This can be helpful, for example:

- when an application error has destroyed a table and this has not been detected during several archiving periods.
- if some table from a specific period, such as year end closing, would be needed for comparison reasons, you might want to keep an extra archive set of that period, to be able to restore individual tables. No additional backups would have to be made.

Starting from that older archive, a table can be restored individually with forward recovery and this can be stopped at exactly the point in time that you decide, where the data reflects the correct information that you need.

Backup Occasions

In addition to regular backups (see “Archive Frequency” on page 62), consider to include backups in your strategy:

- Before and after an add or delete dbextent
- Before and after an add dbspace
- After dbspaces have been moved from storage pools, or other major layout changes have been applied to your database
- After having loaded a large quantity of data or having applied major updates in single user mode with LOGMODE=N
- Before migrating DB2 to a different level or release
- After installing a new database
- Before installing new programs, such as QMF, in the database
- Before changing the size of the log disk
- Before switching from single to dual log or the other way around
- Before changing the size of the directory
- Before copying the directory to a new extent to implement VMDSS

5.2 Evaluation

Without the Data Restore feature, you might need **duplicate backups** of certain tables or dbspaces, in addition to regular archives taken. There has been no possibility to restore individual tables from a DB2 archive or user archive. To be able to restore individual tables, often a backup with DBSU UNLOAD has been taken additionally. Since it is not predictable which table or dbspace of a backup will be needed for restore, in the worse case a copy of every table would have to be available. So very often a selection of important data was made and only these were separately copied, others were missing. An additional problem lies in the fact that **no forward recovery** can be applied on tables.

With the Data Restore feature, you get the biggest flexibility using either the DB2 ARCHIVE or Data Restore BACKUP for making backups. The Data Restore RESTORE lets you restore a full database or just one storage pool from a complete archive. For information about log recovery after RESTORE of a storage pool, please refer to 2.5, “Log Recovery” on page 13. The Data Restore RELOAD gives you the possibility to reload any table from a complete archive, and there is no need to make additional copies of tables just for backup. Therefore, it is recommended to spend the resources on an **improved archive process** of all data rather than on making duplicate backups of specific tables.

There are still occasions where you will need to unload and reload tables:

- When moving tables between storage pools
- When copying tables to a different database
- When modifying characteristics of a table or dbspace

For these kinds of operations there are several alternatives from which to choose depending on which alternative with its capabilities fits your needs best.

5.2.1 Archive Advantages

DB2 ARCHIVE and Data Restore BACKUP offer you the following recovery choices:

- Forward Recovery

  As with DB2 restore, forward recovery is possible with Data Restore RESTORE, if LOGMODE=L or LOGMODE=A is used.

- Recovery with filtered log

  In case of a DBSS failure, the database might, for example, abnormally end during startup indicating the failing LUW. During startup you can specify to omit a specific LUW. This holds for:

  – DB2 restore

  – Restart after a Data Restore RESTORE of a complete database

  – Restart after a Data Restore RESTORE of a storage pool

  – Restart having restored any user archive of the complete database

- Data Restore RELOAD of a single table

  As described above, this relieves you from making additional unloads (duplicate backups) of single tables.

- Data Restore RELOAD with forward recovery of a table, even to point in time.

  With Data Restore LISTLOG you can easily find the statement that corrupted your table and omit that (and the following commands) during recovery.

5.2.2 Archive Weaknesses

There are very few disadvantages of DB2 ARCHIVE and Data Restore BACKUP, depending on what you compare them to.

- Compared to **backup** of **a table only**, it is a complete backup, which takes more time than just an unload of a table.

  But normally, unloads have been made **in addition** to complete backups, so that these times in fact were added.

- **User archives**, for example with DDR, might have included the necessary files to recover a database on another system or after exchanging all dbextents.

  Therefore, in addition to offline DB2 ARCHIVE or Data Restore BACKUP, you should **save the database definition files** as described in chapter 4.2.1, “Recover from System Failure” on page 52.

5.2.3 Comparison between Data Restore BACKUP and DB2 ARCHIVE

The differences between DB2 ARCHIVE and Data Restore BACKUP are the following:

DB2 ARCHIVE

- Archive can be made **online**, without reducing the availability of the database.

  For disaster recovery on a different system, it is good to have an archive available which was taken offline, or online in LOGMODE=L, because these are consistent. Otherwise you need to be able to apply forward log recovery to get consistent. The same holds when you want to use an older archive

than the latest, as described in chapter 4.2.1, “Recover from System Failure” on page 52.

- Data Restore TRANSLATE immediately or later.

  You can translate every DB2 archive immediately after having taken it and keep the work files.

  A Data Restore RELOAD can also be made without translation, needing more time; the translation is made internally, reading the tape twice. But this takes less time than TRANSLATE and RELOAD from translated archive.

  For Data Restore RESTORE a TRANSLATE would be required; you can omit translation when restoring the database using the standard DB2 restore procedure.

Data Restore BACKUP

- You do **not** need to **translate** the archive for:
  - Data Restore RESTORE of a complete database
  - Data Restore RESTORE of a storage pool
  - RELOAD of an individual table

    All of these operations can be performed directly from a Data Restore archive.
- You can make archives **to disk**.

  Considering the restrictions about the size of the output disk related to the size of the database, this applies only for small and medium databases (see chapter 2.6, “Data Restore BACKUP” on page 13).

  Given the fact that we measured around double the time for an archive to disk than to tape, this is not a big advantage.
- You can make **dual archives**: to tape, to disk, or mixed.

  For example, you might want to keep one tape copy for table recovery, and send another tape copy to a safe place for disaster recovery.

So the most prominent difference is that a **DB2 archive can be taken online, but might be translated**, which can take twice as long as the archive itself. For disaster recovery in LOGMODE=A, offline archives are safer, as described in chapter 4.2.1, “Recover from System Failure” on page 52 and “Disaster Recovery” on page 64.

5.3 Considerations

The following is a checklist of factors that can influence the planning of your archive and recovery strategy:

- The environment: production, development or test, or is data recreatable from another database or platform?
- The size of the database
- The vitality of data: frequency of updates, amount of updates and frequency of dataload
- The time it takes to apply the logfiles
- The facilities available for unattended (automated) archive

Taking into account the factors from the list above, some decisions should be made how you run your database and how you define your recovery strategy:

- Frequency of reorganization
- Type of archive: DB2, Data Restore or user archive
- Archive to disk or tape
- Archive online or offline
- Frequency of archive
- Number of backup cycles to be kept
- Single or dual log
- LOGMODE parameter
- Size of the log disk
- Checkpoint interval
- Disaster recovery

The following paragraphs outline considerations related to selected items of the lists above.

Type of Data

Some differences in backup could apply if the main purpose of the data residing in the database is statistics or running queries to determine trends, if the data is imported from somewhere else.

You can consider to reduce the frequency of backup when data is just history data and has very few updates.

You can consider even to omit regular backups, when the data is unimportant or reproducible:

- When data is copied from another data source, for example for statistics for the purpose of running queries to determine trends, when the data originates partially or completely from a production database.
- Development or test database

Database Size

The backup strategy for a very large database is completely different from that for a small database. For a large one, online archives could lead to a checkpoint during the archive process preventing the users from work, so offline archives might be better suited, and if they take too long, it might even be not possible to perform an archive every day. For a small database, daily online archives can be made, even a Data Restore BACKUP to disk could be realistic, or just creating an image copy on a different set of disks; DDR disk to disk is very quick and can be an intermediate step before copying to tape.

Data Manipulation: Frequent Updates

If the data is very important and updated often, it is recommended to increase the size of the log disk, so that the interval between archives or log archives could be increased. For LOGMODE=L you should consider that for frequent updates it can take a very long time to apply forward recovery.

Type of Archive

Before the archive enhancements in SQL/DS V3R5 a tendency was to take *user archives*. Now that DB2 archive performance is improved, it gives you the possibility to choose between DB2 ARCHIVE and Data Restore BACKUP. Data Restore RELOAD of individual tables is possible in both cases, with the only difference that DB2 archive tapes have to be read twice if not translated to a Data Restore format. Data Restore RESTORE of a storage pool requires translation or a Data Restore BACKUP.

Archive Medium

For Data Restore BACKUP, user archives and (in VM) log archives, you have to decide whether to make them to disk or tape. To disk might be faster when using DDR or similar tools; with BACKUP we measured a better performance to tape. For the purpose of disaster recovery, backup to tape is required. For DB2 ARCHIVE only tape is supported.

Archive Frequency

The interval between archives and log archives depends on the frequency of data modifications in your database and the allocated size for the log disk. This should influence your decision which logmode you choose. With log archives (LOGMODE=L) often the interval between archives is increased. If you choose this, consider that for frequent updates it can take a very long time to apply forward recovery. Therefore, it is a good strategy to try to reduce the interval between archives. If time allows and if your archive procedure is fully or almost fully automated, this can be easy.

Additionally, make archives at certain points in time as described in chapter “Backup Occasions” on page 57.

Archive Online or Offline

Using DB2 ARCHIVE, you can choose to perform online because its speed has improved in SQL/DS V3R5. For disaster recovery and for restoring a back level archive in LOGMODE=A, you might want to make offline archives from time to time, see “Disaster Recovery” on page 64. For Data Restore BACKUP and user archives you have no choice; they must be taken offline.

Directory Verification

Whenever you perform an SQLEND, do it with the parameter DVERIFY to verify the directory. After larger updates of the database you might want to take the database manager offline just for the purpose of the database verification, as described in 4.1.2.2, “Directory Disk Failure” on page 50.

Archive Sets

You should decide how many archive sets will be kept, before re-using the tapes. The number of tapes belonging to each set could be the main factor that influences this number of archive sets. As mentioned before, there might be some need to restore an older archive set due to a problem already present on the latest archive. If LOGMODE=L is used, all intermediate log files should also be available for this restore.

For example, if you are running your database in LOGMODE=L and you take archives every other day and log archives daily, you might want to keep archives and log archives for one week, and additionally the first (offline) archive of every month for one year.

For LOGMODE=A, an online archive can only be used if this is the newest one because online archives need the current log to become consistent. When you restore an older archive, you must choose one that has been taken offline.

For more information about log recovery after RESTORE of a storage pool, please refer to 2.5, “Log Recovery” on page 13.

Logging

The log archive takes less time than a full database archive. In VM it is possible to create log archives on disk.

There are some advantages creating log archives on disk instead of tape:

- No tape mounting procedure is involved either in archive or in restore of the log
- All log archive files can be available on the same disk, so the identification is easier
- Log archives to disk are faster than to tape

You must decide whether you need the log archives even in case of disaster recovery; if so, you need a tape copy.

When you update large amounts of data with DATALOAD or RELOAD in multiple user mode, log consumption can be high. This can be avoided by running DBSU in single user mode with LOGMODE=N. Or, for DATALOAD, you can specify the COMMITCOUNT option together with LOGMODE=Y in multiple user mode.

Dual Log

The database manager requires at least one log disk and supports dual logging. Dual logging is recommended when high availability of the database is required because it helps to recover from log disk failures. Especially in VSE, where there is no additional copy of the log history file, this option is recommended.
In **VM**, make sure that the two log disks and the 191 disk of the database are on different DASDs.
In **VSE**, make sure that the two log extents (clusters) not only reside on different volumes (DASDs), but also in different VSAM catalogs in case of a catalog error.

LOGMODE

It should be decided, whether LOGMODE=A or LOGMODE=L will be used, because if a restore of the complete database is requested, it can take some time to apply all log files starting from the last archive.

In LOGMODE=L an online archive can be used for disaster recovery or to restore an older level of archive; in LOGMODE=A, for these two cases only offline archives can be used, as described in chapter 4.2.1, “Recover from System Failure” on page 52.

Log Size

A useful approach to calculate the size of your log is to estimate the percentage of data that will be generated, deleted, and changed over one archive period. You can estimate the log space requirement with the following calculation:

```
logsize estimate = (percentage generated
                    + percentage deleted
                    + percentage changed x 2) x database size
```

Checkpoint Interval

The database manager takes periodic checkpoints depending on the startup parameter CHKINTVL. This parameter defines how many log pages are to be written before DB2 automatically takes a checkpoint. A checkpoint record is written to the log to synchronize the log with the state of the database. The checkpoint parameter CHKINTVL controls the duration between checkpoints. Many installations find that the optimum CHKINTVL setting is between 50 and 300. Installations with small databases tend to be in the upper end of that range. Try to adjust the startup parameter CHKINTVL to have one checkpoint taken every 15 minutes.

If DB2 archives are taken **online**, you normally **choose times of low activity**. However, because your users want to continue to work, the CHKINTVL must be chosen large enough to avoid that during the archive another checkpoint (than the one at the beginning of the archive) will be requested; when that happens, your users will be forced to wait until the archive is finished.

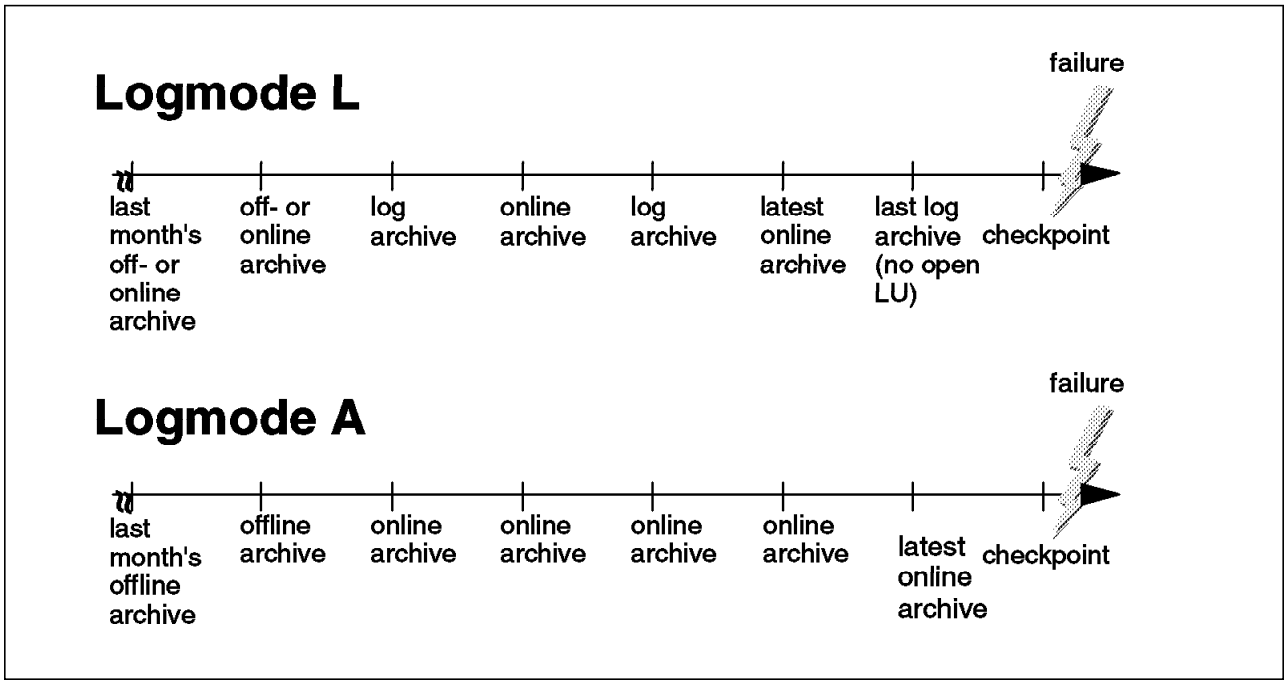


Figure 44. Sample Event Timing

Disaster Recovery

Depending on your recovery strategy for your complete system, you will have to determine additional archiving steps. If your recovery procedure includes the ability to move your complete operating environment to a different system, for

example in an IBM Backup and Recovery Center, and you can rely on regular backups taken, you need not worry about database definition files.

If this is not planned but your database should be able to be restarted on a different system, for example, with remote applications accessing your database, you need to keep an actual copy of your **database definition files** together with your archives.

All archives, log archives and files must reside **on tapes** for use on a different system.

When you are running in LOGMODE=A and performing regular **online** archives, you should consider to make additional offline archives (of any type) with a lower frequency, which might be acceptable for disaster recovery (see “Fallback Time”). If a lower frequency even for the case of a disaster is not acceptable, you can run your database with LOGMODE=L, make log archives to tape, and after each one:

- in **VM** copy the log history to tape
- in **VSE** take your database offline to copy the log file to tape.

For more details, see chapter 4.2.1, “Recover from System Failure” on page 52.

Fallback Time

When the database is restored after a failure, it represents the contents of a certain point in time backward from the failure; ideally this would be the moment immediately before the failure.

Depending on:

- the type of failure you consider
- the fallback time that your organization is willing to accept for that case
- the regular effort you can spend for backup
- the interruptions allowed to the availability of the database

you must choose the archive frequency, the logmode, and whether you perform archives online or offline.

Consider the following sequence of events, as shown in Figure 44 on page 64.

- In case the system suddenly came down, in both LOGMODEs A and L, the database falls back to the point in time of the last checkpoint. Using the log, incomplete LUWs are rolled back, and completed LUWs are re-created completely and committed.
- In case of a DASD failure or an immediately detected database corruption,
  - if LOGMODE=L, you can install the last archive taken, and then perform forward log recovery using log archives and the current log.
  - if LOGMODE=A, you can install the last archive and perform forward log recovery from the current log.
- In case a database corruption had occurred some time back and is detected at the indicated point in time of failure,
  - if LOGMODE=L, restore from the latest archive from the point in time before the cause, and if you still have all the log archive tapes in between then and now, you can perform forward log recovery with these,

which might take a long time. You can filter the recovery to avoid the damage being repeated.
  - if LOGMODE=A, you have to restore from the latest offline archive before the cause, and no forward log recovery will be possible.
- In case of a disaster, you fall back
  - if you are running in LOGMODE=L, to the (online or offline) archive you install, and additionally forward recovery of all consecutive log archives can be made (although this might take much time), provided you have the log archives without gap and the log history file has the appropriate information. That is depending on your recovery strategy for your complete system.
  - if you are running in LOGMODE=A, to the offline archive you install. No forward log recovery can be made, and online archives can not be chosen because the current log is not available to put the archive into a consistent state.

5.4 Possible Strategies

In this chapter we describe some database management systems with different requirements to help you define possible strategies for your database environment.

5.4.1 User Archives and Unloads

This strategy looks as follows:

User archive + unloads → *failure* **- restore**

restore is either of:

- **database - user restore**
- **storage pool - you have to restore the entire database**
- **table - DBSU RELOAD without log recovery**

This strategy has been widespread, for example using DDR because it used to be quicker than DB2 ARCHIVE until SQL/DS V3R5. In addition to the user archive, the important tables or even all tables were unloaded to be able to fix smaller problems quickly.

You should make sure that the database definition files and the log file are included in your user archive.

With the enhanced capabilities of SQL/DS V3R5 and the availability of the Data Restore feature, it might be advantageous to migrate to one of the following strategies.

5.4.2 Online DB2 Archive and Translation

This strategy looks as follows:

Online DB2 ARCHIVE - Data Restore TRANSLATE → *failure* **- restore**

restore is either of:

- **database - Data Restore RESTORE or a DB2 restore**
- **storage pool - Data Restore RESTORE** of a storage pool from translated DB2 archive
- **table - Data Restore RELOAD** from translated DB2 archive with log recovery

Because of the availability of the database, the DB2 archive is always followed by a translate process, keeping a copy of the work files. This has been chosen because the reload of individual tables with forward recovery is considered to apply and must perform quickly.

If you need to recover one or more storage pools, Data Restore RESTORE of a storage pool is the best choice.

When individual tables need to be recovered, Data Restore RELOAD is the best alternative.

For a complete restore you can choose between a restore from DB2 and Data Restore RESTORE.

Complete log recovery is possible in both cases, RELOAD of a table and RESTORE of the whole database, and you will be prompted for the log archives (if any) according to the information that is available in the log history.

With a lower frequency, any kind of offline archive is made for disaster recovery. For the same purpose, a copy of the database definition files is kept with the archives.

5.4.3 Online DB2 Archive

This strategy looks as follows:

Online DB2 ARCHIVE $\rightarrow$ *failure* **- restore**

restore is either of:

- **database - DB2 restore**
- **storage pool - Translate and then Data Restore RESTORE** of a storage pool, or just **DB2 restore** of the entire database
- **table - Data Restore RELOAD** from DB2 archive with log recovery

Because reload with forward recovery is not requested with ultimate performance, the translate process has been chosen to be bypassed.

Similarly, it was decided for the recovery of a storage pool:

- Either a translation is done and then a Data Restore RESTORE of a storage pool
- or a complete DB2 restore process is made.

When a table needs to be recovered, Data Restore RELOAD can be run from the DB2 archive. The reload of tables is slower, because the tapes are read twice, but this time is accepted because reload is not often requested. A lot of time and effort is saved in comparison to the scenario where a Data Restore TRANSLATE is executed on each archive.

For the case of a complete restore, DB2 restore can be applied, which is equivalent to the slower alternative of first running a Data Restore TRANSLATE and then Data Restore RESTORE.

As in the previous scenario, complete log recovery is possible in both cases, RELOAD of a table and restore of the whole database, and you will be prompted for the log archives (if any), according to the information that is available in the log history.

With a lower frequency, any kind of offline archive is made for disaster recovery. For the same purpose, a copy of the database definition files is kept with the archives.

5.4.4 Data Restore BACKUP

This strategy looks as follows:

Data Restore BACKUP → *failure* **- Data Restore RESTORE**

restore is either of:

- **database - Data Restore restore**
- **storage pool - Data Restore RESTORE** of a storage pool
- **table - Data Restore RELOAD** from Data Restore archive with log recovery

There are three reasons why this strategy can be chosen:

- Putting the database offline for the archive, is not considered a problem, because there is a window where the database is allowed to be offline.
- Taking online archives takes too long. Checkpoints might occur during the archive, and performance degradation cannot be accepted for so long. Although it would be preferable to have the database online, offline archive is chosen.
- For offline archives, Data Restore BACKUP is chosen because no translate process is needed.

If you need to recover one or more storage pools, Data Restore RESTORE of a storage pool is the best alternative.

When individual tables need to be recovered, Data Restore RELOAD is the choice.

For a complete restore you use Data Restore RESTORE.

Complete log recovery is possible in both cases, RELOAD of a table and RESTORE of the whole database, and you will be prompted for the log archives (if any), according to the information that is available in the log history.

For disaster recovery, a copy of the database definition files is kept with the archive.

5.5 Migration Scenarios

This section gives some information about migration scenarios and recommendations how the migration could be performed.

Because the DB2 archive was not the best performer for backup of the DB2 database, many customers use the offline procedure, user archive. To be able to restore individual tables, also additional unload of selected dbspaces was performed.

Now that the Data Restore feature is available and the SQL/DS V3R5 archive has been enhanced, there are some alternatives to be considered.

From online DB2 ARCHIVE:

If you are using the DB2 ARCHIVE **online**, there is no need to migrate. You get the additional advantage of the performance enhancement to the archive process.

If you used to make additional backups in form of UNLOAD, you can omit this. Data Restore RELOAD allows you to RELOAD tables from the archive online, and additionally offers you the ability to apply forward log recovery to the reloaded tables. You can choose to use either the implicit (in Data Restore RELOAD) or explicit (in TRANSLATE) translate process. You can also use the Data Restore RESTORE command to recover a storage pool.

You just have to install the Data Restore feature and become familiar with the Data Restore RELOAD and RESTORE, and optionally TRANSLATE process, as well as LISTLOG and APPLYLOG. See Figure 45.

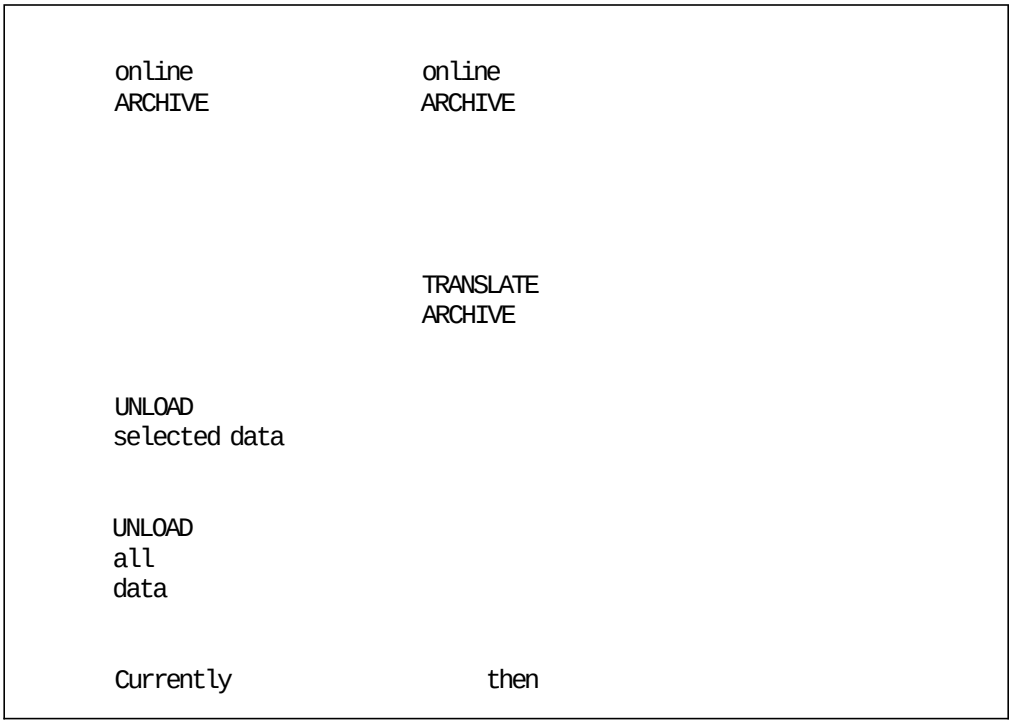


Figure 45. Migration from Online DB2 Archive

From offline DB2 archive:

If DB2 archive was used **offline** with SQLEND ARCHIVE, you can keep this archive option and consider the TRANSLATE, or you can migrate to Data Restore BACKUP to omit the TRANSLATE. You can also switch to online archive, because it now takes less time. This might take a little bit more time than offline, if there are any users running concurrently with the archive, but there is the advantage of keeping the database online. See Figure 46 Choices A, B and C.

If you used to make additional backups in form of UNLOAD, you can omit this. Data Restore RELOAD allows you to RELOAD tables from the archive online, and additionally offers you the ability to apply forward log recovery to the reloaded tables. You can choose to use either the implicit (in Data Restore RELOAD) or explicit (in TRANSLATE) translate process.

You just have to install the Data Restore feature and become familiar with the Data Restore RELOAD and RESTORE, and BACKUP or TRANSLATE process, as well as LISTLOG and APPLYLOG.

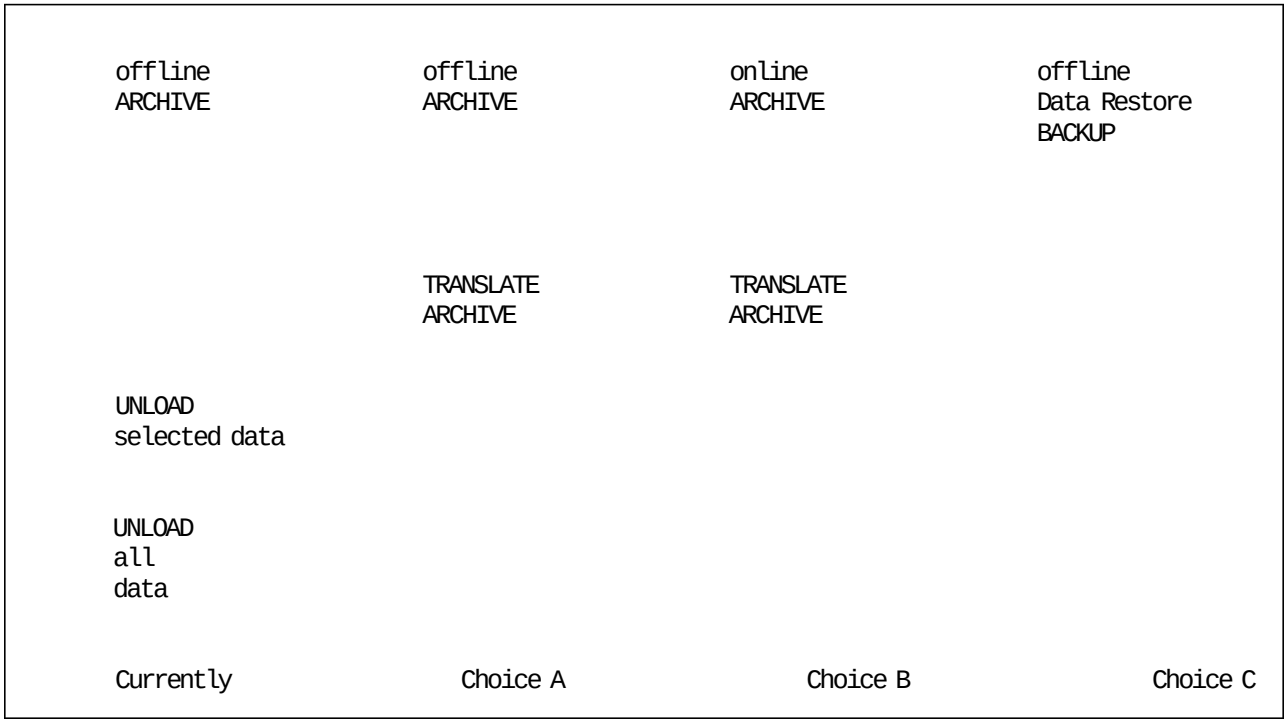


Figure 46. Migration from Offline DB2 Archive

From user archive:

Most of the user archive tools were better performers than the DB2 archive. In addition to this, unload of selected dbspaces was used to have the possibility for individual restore. (See Figure 47 on page 72 “Currently.”) If all or a major part of the dbspaces would be unloaded, this could take some time and resources, although the unload could run when the database was online.

Now you have two choices:

1. Migrate to Data Restore feature BACKUP, which is also performed with the database offline. This might take some more time than your usual archive, (see Figure 47 on page 72 “Choice A”), but has major advantages:
   - No maintenance of the input definitions (to run the user archives) is required. The Data Restore feature just needs access to the production disk (default 195) in VM, or to the database identification job, containing the DLBL statements for the extents in VSE.
   - All storage pools are available for individual restore.
   - All tables are available for individual restore. Thus, additional UNLOAD of selected or all data can be omitted.
   - Reload tables with forward recovery is possible.
2. If the availability of the database is a major requirement, migration to online DB2 ARCHIVE could be considered. The possible disadvantage is the additional translate process, which can be performed at any time. (See Figure 47 on page 72 “Choice B.”)

For the migration to online DB2 archive, the procedures would have to be changed, so the operator command should be executed instead of ending the database, running your user archive, and later possibly doing additional unloads. If you run with LOGMODE=A, you might want to additionally take offline archives at longer intervals for disaster recovery.

If Data Restore feature BACKUP is selected as an alternative for your archive, then there would be no change in the process of ending your database. You just replace your user archive procedure by starting the Data Restore BACKUP.

In both cases, additionally having the database definition files (to be copied only after changes of the database layout) should be considered, as described in chapter 4.2.1, “Recover from System Failure” on page 52, to have the chance to restore and run the database on another system. You might want to check whether these files have been included in your current user archives or not, and whether restore on another system is reflected in your recovery strategy.

You can improve the recovery even further by taking a copy of the log history to tape after each log archive, as described in chapter 5.3, “Considerations” on page 60, ″Disaster Recovery″.

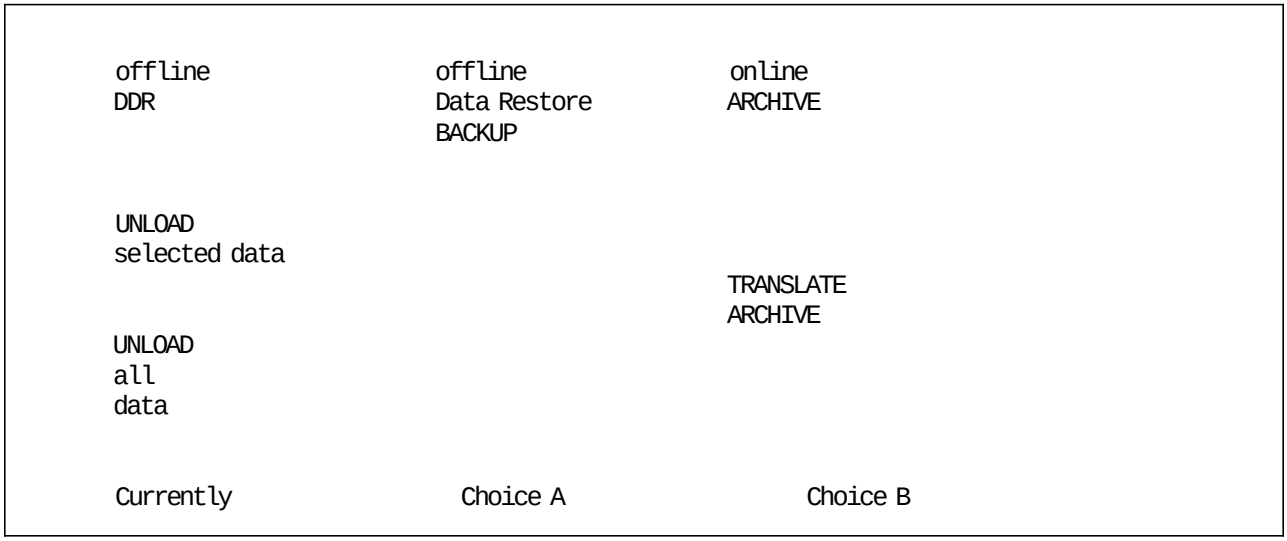


Figure 47. Migration from User Archive

5.6 Feedback

Please let us know about your archiving experiences. This will help us for the future update of this redbook. Since customer's environments are individual, we are interested to get information about your specific situation.

Please tell us your current and your future recovery strategy:

- Kind of archive performed?
- Logmode used?
- Data additionally unloaded?
- Reasons why you do so?
- Experiences made?

Please send your comments together with the evaluation form or directly:

Mail:
Eberhard Lange, ITSO
IBM Laboratory Dept. 3222
Schönaicher Str. 220
71032 Böblingen

Fax: +49-7031-16-1221

Email: elange@vnet.ibm.com

Appendix A. Test Scenarios Using SQL/DS V3R5

This chapter gives you an overview of the test environments that have been run using SQL/DS V3R5 and some figures about the performance of these tests.

A.1 Environment Overview

The following describes the test environment on which these tests have been performed.

Processor

- IBM ES/9000 9221 Model 150

Tape Unit

- IBM model 3480

Operating system

- VM/ESA Version 1 Release 2 Modification 2 (1.2.2)
  with Minidisk Caching switched on.
- VSE/ESA Version 2 Release 1 Modification 2 (2.1.2)

VSE is running on VM.

Database Server

- SQL/DS Version 3 Release 5 Modification 0 (3.5.0)
- Data Restore feature Version 3 Release 5 Modification 0 (3.5.0)

Database Layout

The database was defined with 14 storage pools.

- The data tables loaded under VM and VSE are identical.
- Each table has its own dbspace and its own storage pool.
- All storage pools are large enough. For space reasons not all storage pools have the same size in VM and VSE, but the difference is only in the spare size of the pools and thus is irrelevant for the measured time.

Further details of the database specifications are presented in the according environment.

Sample Table

For the Data Restore feature functions UNLOAD and RELOAD, a table was used with the following definitions:

- dbspace with 4096 pages
- located in storage pool 8
- a total of 2737 datapages and 2223 used pages, 81%
- a total of 1351 indexpages and 228 used pages, 16%

- before the reload, the dbspace was first dropped and acquired, to avoid a storage pool full or log full condition

A.2 VM Environment and Measurements

First some settings and information that remained constant during all the tests that were performed:

- Database server with SHARE RELATIVE 1000 and QUICKDSP OFF
- User for Data Restore feature tests with SHARE RELATIVE 100 and QUICKDSP OFF
- Total number of database pages 144793
- Directory disk (BDISK) 80 cylinders on 3380
- Tape unit is 3480 without hardware compaction feature
- FILEDEF ARIARCH TAP1 SL (RECFM FB BLOCK 28672 LRECL 4096
- FILEDEF LARCHIV TAP2 SL (RECFM FB BLOCK 28672 LRECL 4096
- FILEDEF ARCHIV TAP1 SL (RECFM VB BLOCK 32760

POOL	PAGES	USED	FREE	VOLUME	CYL
BDISK				IS2LS1	80
LOG				MM45R1	40
1	11970	1864	10106	MM45R3	80
2	11970	16	11954	MM45R0	80
3	3705	395	3310	LS45R5	25
4	3705	393	3312	LS45R5	25
5	3705	14	3691	LS45R5	25
6	7467	712	6755	LS45R5	50
6	7467	713	6754	LS45R6	50
6	7467	699	6768	LS45R7	50
6	22401	2124	20277	TOTAL	150
7	11970	0	11970	LS45R6	80
8	5985	3291	2694	LS45R5	40
9	44916	14359	30557	LS45R5	300
9	44916	14351	30565	LS45R6	300
9	44916	14349	30567	LS45R7	300
9	134748	43059	91689	TOTAL	900
10	3705	3	3702	LS45R5	25
11	7467	2	7465	LS45R5	50
11	7467	0	7467	LS45R6	50
11	7467	0	7467	LS45R7	50
11	22401	2	22399	TOTAL	150
12	7467	3044	4423	LS45R5	50
12	7467	3002	4465	LS45R6	50
12	7467	3045	4422	LS45R7	50
12	22401	9091	13310	TOTAL	150
13	7467	4842	2625	LS45R5	50
13	7467	4844	2623	LS45R6	50
13	7467	4825	2642	LS45R7	50
13	22401	14511	7890	TOTAL	150
14	44916	23345	21571	LS45R5	300
14	44916	23341	21575	LS45R6	300
14	44916	23345	21571	LS45R7	300
14	134748	70031	64717	TOTAL	900
TOTAL	415815	144794	271021		2900

Figure 48. Database Configuration on VM

The following table gives some information about the tests that have been performed in VM.

Elapsed Time The elapsed time is the time between the start and end of each test.

I/O The I/O is the difference between the number of I/O, given by the command *CP IND USER **, at the start and at the end of the test. For the DB2 ARCHIVE and RESTORE operation these are the numbers of I/O requested by the

database. In all other tests these are the numbers of I/O requested by the user ID that is executing the Data Restore feature commands.

Media Media indicates the number of tapes or number of cylinders that were used as input or produced as output by this test.

Table 6. Test Results VM

Function	Elapsed Time	I/O	Media
SQL/DS ARCHIVE	9:11	31469	Out 3 tapes
SQL/DS restore from SQL/DS archive	51:33	221982	In 3 tapes
TRANSLATE SQL/DS archive to Tape	22:28	68356	In 3 tapes Out 4 tapes
Data Restore BACKUP to tape (Data Restore archive)	11:29	22643	Out 3 tapes
Data Restore BACKUP to disk (Data Restore archive)	22:57	78275	Out 1000 cyl
Data Restore RESTORE from Data Restore archive tape	51:03	221302	In 3 tapes
Data Restore RESTORE from Data Restore archive disk	49:50	228809	In 1000 cyl
Data Restore RESTORE from Translated SQL/DS archive tape	52:22	221384	In 4 tapes
Data Restore RELOAD from Data Restore archive tape	8:19	21698	In 3 tapes
Data Restore RELOAD from SQL/DS archive tape	18:07	46906	In 3 tapes
DBSU UNLOAD to disk	1:06	2228	Out 17 cyl
DBSU RELOAD from DBSU unload disk	2:25	8256	In 17 cyl
Data Restore UNLOAD to disk	0:43	1328	Out 16 cyl
Data Restore RELOAD from Data Restore unload disk	4:18	1208	In 16 cyl

A.3 VSE Environment and Measurements

First some settings and information that remained constant during all the tests that were performed:

- VSE machine with SHARE RELATIVE 100 and QUICKDSP ON
- Total number of database pages 114212
- Directory disk (BDISK) 80 cylinders on 9345
- Tape unit is 3480 without hardware compaction feature

```
POOL    PAGES       USED      FREE       CYL
BDISK                                    80
LOG                                      80
 1     11970        222      11748       80
 2     11970         16      11954       80
 3      3705        372       3333       25
 4      3705        393       3312       25
 5       741         14        727        5
 6      7467       2124       5343       50
 7     11970          0      11970       80
 8      5985       2452       3533       40
 9     89946      43059      46887      600
10       741          3        738        5
11      3705          2       3703       25
12     22458      10226      12232      150
13     22458      14511       7947      150
14     89946      40818      49128      600

TOTAL 296767     114212     182555     2075
```

Figure 49. Database Configuration on VSE

```
// PROC CAT=¢SQLPCAT¢
// DLBL IJSYSUC,¢SQLPCAT.USER.CATALOG¢,,VSAM
* ********************************************************
* SQLVSE02: SQL/DS DATABASE IDENTIFICATION
* ********************************************************
// DLBL BDISK,¢SQLVSE02.BDISK.SQLDIR80¢,,VSAM,CAT=&CAT
// DLBL LOGDSK1,¢SQLVSE02.LOGDSK1.SQLLOG¢,,VSAM,CAT=&CAT
// DLBL DDSK1,¢SQLVSE02.DDSK1.POOL1¢,,VSAM,CAT=&CAT
// DLBL DDSK2,¢SQLVSE02.DDSK2.POOL2¢,,VSAM,CAT=&CAT
// DLBL DDSK3,¢SQLVSE02.DDSK3.POOL3¢,,VSAM,CAT=&CAT
// DLBL DDSK4,¢SQLVSE02.DDSK4.POOL4¢,,VSAM,CAT=&CAT
// DLBL DDSK5,¢SQLVSE02.DDSK5.POOL5¢,,VSAM,CAT=&CAT
// DLBL DDSK6,¢SQLVSE02.DDSK6.POOL6¢,,VSAM,CAT=&CAT
// DLBL DDSK7,¢SQLVSE02.DDSK7.POOL7¢,,VSAM,CAT=&CAT
// DLBL DDSK8,¢SQLVSE02.DDSK8.POOL8¢,,VSAM,CAT=&CAT
// DLBL DDSK9,¢SQLVSE02.DDSK9.POOL9¢,,VSAM,CAT=&CAT
// DLBL DDSK10,¢SQLVSE02.DDSK10.POOL10¢,,VSAM,CAT=&CAT
// DLBL DDSK11,¢SQLVSE02.DDSK11.POOL11¢,,VSAM,CAT=&CAT
// DLBL DDSK12,¢SQLVSE02.DDSK12.POOL12¢,,VSAM,CAT=&CAT
// DLBL DDSK13,¢SQLVSE02.DDSK13.POOL13¢,,VSAM,CAT=&CAT
// DLBL DDSK14,¢SQLVSE02.DDSK14.POOL14¢,,VSAM,CAT=&CAT
```

Figure 50. DLBL for Dbextents (SQLVSE02.PROC)

The following table gives some information about the tests that have been performed in VSE.

Elapsed Time The elapsed time is the time between the start and end of each test.

CPU This indicates the average percentage of CPU usage for the user ID where VSE was running.

I/O Rate I/O Rate is the number of I/O per second requested by the user ID where VSE was running.
CPU and I/O Rate are figures based on the output of MONWRITE, the monitor running in VM, and presented as reports using VMPRF.

Media Media indicates the number of tapes or number of cylinders that were used as input or produced as output by this test.

Table 7. Test Results VSE

Function	Elapsed Time	CPU	I/O Rate	Media
SQL/DS ARCHIVE	9:46	9.6	8.94	Out 3 tapes
SQL/DS restore from SQL/DS archive STARTUP=F	37:10	14.6	71.65	In 3 tapes
SQL/DS restore from SQL/DS archive STARTUP=R (formatting all)	1:51:28 hr	6.4	78.79	In 3 tapes
TRANSLATE SQL/DS archive to tape	20:56	4.6	2.18	In 3 tapes Out 3 tapes
TRANSLATE SQL/DS archive to disk	28:31	4.1	12.5	In 3 tapes Out 918 cyl
Data Restore BACKUP to tape (Data Restore archive)	9:25	7.4	10.7	Out 3 tapes
Data Restore BACKUP to disk (Data Restore archive)	22:46	4.0	18.17	Out 918 cyl
Data Restore BACKUP Dual	25:20	4.6	16.37	Out 3 tapes Out 918 cyl
Data Restore RESTORE from Data Restore archive tape	48:03	6.4	79.7	In 3 tapes
Data Restore RESTORE from Data Restore archive disk	54:33	5.7	76.42	In 918 cyl
Data Restore RESTORE from Translated SQL/DS archive tape	48:29	6.2	79.31	In 3 tapes
Data Restore RELOAD from Data Restore archive tape	9:01	14.1	10.22	In 3 tapes
Data Restore UNLOAD to disk	00:49	2.7	25.49	Out 16 cyl
Data Restore RELOAD from Data Restore unload disk	03:29	27.2	26.28	In 16 cyl
DBSU UNLOAD to tape	01:11	18.2	0.76	Out 1 tape
DBSU RELOAD from tape	02:24	16.1	31.66	In 1 tape

Appendix B. Test Scenarios Using DB2 Server for VSE & VM V5R1

This chapter gives you an overview of the test environments that have been run using DB2 Server for VSE & VM V5R1 and figures about the performance of these tests. We also executed several tests using SQL/DS V3R5 to get some performance comparison on the same processor.

B.1 Environment Overview

The following describes the test environment on which these tests have been performed.

Processor

- IBM ES/9000 9672 Model R72

Tape Unit

- IBM 3490E

Operating System

- VM/ESA Version 2 Release 1
  with Minidisk Caching switched on.
- VSE/ESA Version 2 Release 2

VSE is running on VM.

Database Server

- For SQL/DS V3R5 tests:
  - SQL/DS Version 3 Release 5 Modification 0 (3.5.0)
  - Data Restore feature Version 3 Release 5 Modification 0 (3.5.0)
- For DB2 Server for VSE & VM V5R1 tests:
  - DB2 Server for VSE & VM V5R1
  - Data Restore feature Version 5 Release 1

Database Layout

The database was defined with 14 storage pools.

- The data tables loaded under VM and VSE are identical.
- Each table has its own dbspace and its own storage pool.
- All storage pools are large enough. For space reasons not all storage pools have the same size in VM and VSE, but the difference is only in the spare size of the pools and thus is irrelevant for the measured time.

Further details of the database specifications are presented in the according environment.

Sample Table

For the Data Restore feature functions UNLOAD and RELOAD, a table was used with the following definitions:

- dbspace with 4096 pages
- located in storage pool 8
- a total of 2737 datapages and 2223 used pages, 81%
- a total of 1351 indexpages and 228 used pages, 16%
- before the reload, the dbspace was first dropped and acquired, to avoid a storage pool full or log full condition

B.2 VM Environment and Measurements for SQL/DS V3R5

First some settings and information that remained constant during all the tests that were performed:

- Database server with SHARE RELATIVE 1000 and QUICKDSP OFF
- User for Data Restore feature tests with SHARE RELATIVE 100 and QUICKDSP OFF
- Total number of used database pages 144297
- Directory disk (BDISK) 80 cylinders on RAMAC (logically 3380)
- Tape unit is 3490E
- FILEDEF ARIARCH TAP1 SL (RECFM FB BLOCK 28672 LRECL 4096
- FILEDEF LARCHIV TAP2 SL (RECFM FB BLOCK 28672 LRECL 4096
- FILEDEF ARCHIV TAP1 SL (RECFM VB BLOCK 32760

POOL	PAGES	USED	FREE		CYL
BDISK					80
LOG					40
1	11970	1991	9979		80
2	11970	52	11958		80
3	3705	270	3435		25
4	3705	393	3312		25
5	3705	14	3691		25
6	7467	785	6682		50
6	7467	761	6706		50
6	7467	767	6700		50
6	22401	2313	20088	TOTAL	150
7	11970	0	11970		80
8	5985	2562	3423		40
9	44916	14359	30557		300
9	44916	14351	30565		300
9	44916	14349	30567		300
9	134748	43059	91689	TOTAL	900
10	3705	5	3700		25
11	7467	2	7465		50
11	7467	0	7467		50
11	7467	0	7467		50
11	22401	2	22399	TOTAL	150
12	7467	3053	4414		50
12	7467	3021	4446		50
12	7467	3017	4450		50
12	22401	9091	13310	TOTAL	150
13	7467	4830	2637		50
13	7467	4843	2624		50
13	7467	4838	2629		50
13	22401	14511	7890	TOTAL	150
14	44916	23354	21562		300
14	44916	23348	21568		300
14	44916	23329	21587		300
14	134748	70031	64717	TOTAL	900
TOTAL	415815	144297	271518		2900

Figure 51. Database Configuration on VM for SQL/DS V3R5

The following table gives some information about the tests that have been performed in VM.

Elapsed Time The elapsed time is the time between the start and end of each test.

The results of the tests may vary depending on the CPU usage, concurrency and external factors.

CPU Time This indicates the CPU time for the user ID where the test was running.

I/O The I/O is the difference between the number of I/O, given by the command *CP IND USER **, at the start and at the end of the test. For the DB2 ARCHIVE and RESTORE operation these are the numbers of I/O requested by the database. In all other tests these are the numbers of I/O requested by the user ID that is executing the Data Restore feature commands.

Media Media indicates the number of tapes that were used as input or produced as output by this test.

Table 8. Test Results for SQL/DS V3R5 in VM

Function	Elapsed Time	CPU Time	I/O	Media
Data Restore BACKUP to tape	9:41	0:10	26624	In 1 tape
Data Restore RESTORE from Data Restore archive tape	30:42	0:31	219571	In 1 tape
Data Restore RELOAD from Data Restore archive tape	8:32	0:08	21409	In 1 tape

B.3 VM Environment and Measurements for DB2 Server for VM V5R1

First some settings and information that remained constant during all the tests that were performed:

- Database server with SHARE RELATIVE 1000 and QUICKDSP OFF
- User for Data Restore feature tests with SHARE RELATIVE 100 and QUICKDSP OFF
- Total number of used database pages 143955
- Directory disk (BDISK) 80 cylinders on RAMAC (logically 3380)
- Tape unit is 3490E
- FILEDEF ARIARCH TAP1 SL (RECFM FB BLOCK 28672 LRECL 4096
- FILEDEF LARCHIV TAP2 SL (RECFM FB BLOCK 28672 LRECL 4096
- FILEDEF ARCHIV TAP1 SL (RECFM VB BLOCK 32760

POOL	PAGES	USED	FREE		CYL
BDISK					80
LOG					40
1	11970	1647	10323		80
2	11970	57	11913		80
3	3705	270	3435		25
4	3705	393	3312		25
5	3705	14	3691		25
6	7467	785	6682		50
6	7467	761	6706		50
6	7467	767	6700		50
6	22401	2313	20088	TOTAL	150
7	11970	0	11970		80
8	5985	2562	3423		40
9	44916	14359	30557		300
9	44916	14351	30565		300
9	44916	14349	30567		300
9	134748	43059	91689	TOTAL	900
10	3705	5	3700		25
11	7467	2	7465		50
11	7467	0	7467		50
11	7467	0	7467		50
11	22401	2	22399	TOTAL	150
12	7467	3053	4414		50
12	7467	3021	4446		50
12	7467	3017	4450		50
12	22401	9091	13310	TOTAL	150
13	7467	4830	2637		50
13	7467	4843	2624		50
13	7467	4838	2629		50
13	22401	14511	7890	TOTAL	150
14	44916	23354	21562		300
14	44916	23348	21568		300
14	44916	23329	21587		300
14	134748	70031	64717	TOTAL	900
TOTAL	415815	143955	271860		2900

Figure 52. Database Configuration for DB2 Server for VM V5R1

The following table gives some information about the tests that have been performed in VM. For a description of **Elapsed Time**, **CPU Time**, **I/O** and **Media**, please refer to B.2, “VM Environment and Measurements for SQL/DS V3R5” on page 80.

Table 9. Test Results for DB2 Server for VM V5R1

Function	Elapsed Time	CPU Time	I/O	Media
Data Restore BACKUP to tape	9:28	0:10	26852	In 1 tape
Data Restore RESTORE from Data Restore archive tape	28:40	0:29	221060	In 1 tape
Data Restore RESTORE of one Storage Pool (Storage Pool number 8)	8:21	0:08	30738	In 1 tape
Data Restore RELOAD from Data Restore archive tape	8:53	0:09	21510	In 1 tape

B.4 VSE Environment and Measurements for SQL/DS V3R5

First some settings and information that remained constant during all the tests that were performed:

- VSE machine with SHARE RELATIVE 100 and QUICKDSP ON
- Total number of database pages 114212
- Directory disk (BDISK) 80 cylinders on RAMAC (logically 3380)
- Tape unit is 3490E

POOL	PAGES	USED	FREE	CYL
BDISK				80
LOG				80
1	11970	309	11661	80
2	11970	97	11873	80
3	3705	372	3333	25
4	3705	393	3312	25
5	741	14	727	5
6	7467	2124	5343	50
7	11970	0	11970	80
8	5985	2452	3533	40
9	89946	43059	46887	600
10	741	3	738	5
11	3705	2	3703	25
12	22458	10226	12232	150
13	22458	14511	7947	150
14	89946	40818	49128	600
TOTAL	296767	114380	182387	2075

Figure 53. Database Configuration on VSE

```
// PROC CAT=¢SQLPCAT¢
// DLBL IJSYSUC,¢SQLPCAT.USER.CATALOG¢,,VSAM
* ******************************************************
* SQLVSE02: SQL/DS DATABASE IDENTIFICATION
* ******************************************************
// DLBL BDISK,¢SQLVSE02.BDISK.SQLDIR80¢,,VSAM,CAT=&CAT
// DLBL LOGDSK1,¢SQLVSE02.LOGDSK1.SQLLOG¢,,VSAM,CAT=&CAT
// DLBL DDSK1,¢SQLVSE02.DDSK1.POOL1¢,,VSAM,CAT=&CAT
// DLBL DDSK2,¢SQLVSE02.DDSK2.POOL2¢,,VSAM,CAT=&CAT
// DLBL DDSK3,¢SQLVSE02.DDSK3.POOL3¢,,VSAM,CAT=&CAT
// DLBL DDSK4,¢SQLVSE02.DDSK4.POOL4¢,,VSAM,CAT=&CAT
// DLBL DDSK5,¢SQLVSE02.DDSK5.POOL5¢,,VSAM,CAT=&CAT
// DLBL DDSK6,¢SQLVSE02.DDSK6.POOL6¢,,VSAM,CAT=&CAT
// DLBL DDSK7,¢SQLVSE02.DDSK7.POOL7¢,,VSAM,CAT=&CAT
// DLBL DDSK8,¢SQLVSE02.DDSK8.POOL8¢,,VSAM,CAT=&CAT
// DLBL DDSK9,¢SQLVSE02.DDSK9.POOL9¢,,VSAM,CAT=&CAT
// DLBL DDSK10,¢SQLVSE02.DDSK10.POOL10¢,,VSAM,CAT=&CAT
// DLBL DDSK11,¢SQLVSE02.DDSK11.POOL11¢,,VSAM,CAT=&CAT
// DLBL DDSK12,¢SQLVSE02.DDSK12.POOL12¢,,VSAM,CAT=&CAT
// DLBL DDSK13,¢SQLVSE02.DDSK13.POOL13¢,,VSAM,CAT=&CAT
// DLBL DDSK14,¢SQLVSE02.DDSK14.POOL14¢,,VSAM,CAT=&CAT
```

Figure 54. DLBL for Dbextents (SQLVSE02.PROC)

The following table gives some information about the tests that have been performed in VSE.

Elapsed Time The elapsed time is the time between the start and end of each test.

The results of the tests may vary depending on the CPU usage, concurrency and external factors.

CPU Time This indicates the CPU time for the user ID where VSE was running.

I/O Number of I/O during the test execution.

Media Media indicates the number of tapes that were used as input or produced as output by this test.

Table 10. Test Results for SQL/DS V3R5 for VSE

Function	Elapsed Time	CPU Time	I/O	Media
Data Restore BACKUP to tape	7:41	0:53	23588	In 1 tape
Data Restore RESTORE from Data Restore archive tape	22:47	3:10	252105	In 1 tape
Data Restore RELOAD from Data Restore archive tape	6:51	2:14	28764	In 1 tape
DBSU RELOAD from tape	1:56	0:42	8634	In 1 tape

B.5 VSE Environment and Measurements for DB2 Server for VSE V5R1

First some settings and information that remained constant during all the tests that were performed:

- VSE machine with SHARE RELATIVE 100 and QUICKDSP ON
- Total number of database pages 114212
- Directory disk (BDISK) 80 cylinders on RAMAC (logically 3380)
- Tape unit is 3490E

The database configuration and the DLBL for dbextents in DB2 Server for VSE V5R1 are described in Figure 53 on page 84 and Figure 54 on page 85 respectively.

The following table gives some information about the tests that have been performed in VSE. For a description of **Elapsed Time**, **CPU Time**, **I/O** and **Media**, please refer to B.4, “VSE Environment and Measurements for SQL/DS V3R5” on page 84.

Table 11. Test Results for DB2 Server for VSE V5R1

Function	Elapsed Time	CPU Time	I/O	Media
Data Restore BACKUP to tape	7:37	0:52	23849	In 1 tape
Data Restore RESTORE from Data Restore archive tape	23:19	3:12	252534	In 1 tape
Data Restore RESTORE of one Storage Pool (Storage Pool number 8)	5:39	0:32	28860	In 1 tape
Data Restore RELOAD from Data Restore archive tape	6:55	2:15	30843	In 1 tape
DBSU RELOAD from tape	1:25	0:41	6519	In 1 tape

Appendix C. Examples

This chapter shows some procedures that have been performed on our test system to illustrate the usage of the Data Restore feature.

C.1 BACKUP Example in VSE

```
* $$ JOB JNM=DRFBACKP,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFBCKUP                BACKUP DATABASE WITH DRF TO TAPE
// LIBDEF *,SEARCH=(PRD2.SQL350,PRD2.RCV350)
// EXEC PROC=SQLVSE02
// TLBL      ARCHIV
// ASSGN     SYS006,180
// MTC       REW,SYS006
// EXEC XTS91001,SIZE=AUTO
OPTIONS DEVICE=TAPE
CONTROL DBNAME=SQLVSE02
BACKUP
/*
/&
* $$ EOJ
```

Figure 55. JCL File for Data Restore BACKUP to Tape

C.2 BACKUP Example in VM

```
/*--------------------------------*/
/*-- filedef for output to tape --------*/
¢FILEDEF ARCHIV   TAP1 SL                (RECFM VB BLOCK 32760¢
/*-- filedef for dual output to disk ---*/
¢FILEDEF ARCHIV2  DISK DRF    BACKUP   A (RECFM VB BLOCK 32760¢
¢FILEDEF SYSIN    DISK BACKUP SYSIN    A¢
¢FILEDEF SYSPRINT DISK BACKUP SYSPRINT A¢
¢XTS91001¢
Exit rc
```

Figure 56. EXEC File for Data Restore BACKUP with Dual Backup to Tape and Disk

```
* backup1 to tape, 2 to disk
OPTIONS DEVICE=TAPE DEVICE2=DASD CONFIRM=NO
CONTROL BASE=S35VMDB1
BACKUP
```

Figure 57. SYSIN File for Data Restore BACKUP with Dual Backup to Tape and Disk

C.3 RELOAD Examples in VM

```
SELECT * FROM SQLDBA.PROJECT
PROJNO PROJNAME              DEPTNO RESPEMP PRSTAFF  ....
------ -------------------- ------ ------- -------
AD3100 ADMIN SERVICES        D01    000010    6.50
AD3110 GENERAL ADMIN SYSTEMS D21    000070    6.00
AD3111 PAYROLL PROGRAMMING   D21    000230    2.00
AD3112 PERSONNEL PROGRAMMING D21    000250    1.00
AD3113 ACCOUNT PROGRAMMING   D21    000270    2.00
IF1000 QUERY SERVICES        C01    000030    2.00
IF2000 USER EDUCATION        C01    000030    1.00
MA2100 WELD LINE AUTOMATION  D01    000010   12.00
MA2110 W L PROGRAMMING       D11    000060    9.00
MA2111 W L PROGRAM DESIGN    D11    000220    2.00
MA2112 W L ROBOT DESIGN      D11    000150    3.00
MA2113 W L PROD CONT PROGS   D11    000160    3.00
OP1000 OPERATION SUPPORT     E01    000050    6.00
OP1010 OPERATION             E11    000090    5.00
OP2000 GEN SYSTEMS SERVICES  E01    000050    5.00
OP2010 SYSTEMS SUPPORT       E21    000100    4.00
OP2011 SCP SYSTEMS SUPPORT   E21    000320    1.00
OP2012 APPLICATIONS SUPPORT  E21    000330    1.00
OP2013 DB/DC SUPPORT         E21    000340    1.00
PL2100 WELD LINE PLANNING    B01    000020    1.00
```

Figure 58. SQLDBA.PROJECT before Updates

In Figure 59 and Figure 60 on page 89 the Data Restore archive is read from tape, as well as the log archives.

```
/*-----------------------------------------------------------------*/
/*- Reload procedure with forward Recovery -------------------------*/
/*- INPUT Data Restore Feature BACKUP - Log Archive - Current LOG---*/
/*-----------------------------------------------------------------*/
Address Command
¢FILEDEF ARCHIV   TAP2 SL 1 (RECFM VB BLOCK 32760¢
¢FILEDEF LARCHIV  TAP1 SL 1 (RECFM FB BLOCK 28672 LRECL 4096¢

¢FILEDEF LMBRWRK DISK LMBRWRK DATA A (RECFM FB BLOCK 28672 LRECL 4096¢
¢FILEDEF LMBRLG1 DISK LMBRLG1 DATA A (RECFM VB BLOCK 32760¢
¢FILEDEF LMBRLG2 DISK LMBRLG2 DATA A (RECFM VB BLOCK 32760¢
¢FILEDEF LMBRLG3 DISK LMBRLG3 DATA A (RECFM VB BLOCK 32760¢

¢FILEDEF SYSIN    DISK RELOAD   SYSIN  A¢
¢FILEDEF SYSPRINT DISK RELOAD   SYSPRINT A¢
¢XTS91001¢
Exit rc
```

Figure 59. EXEC File for Data Restore RELOAD from Data Restore Archive with Forward Recovery

```
XTS9-143 OPTIONS RECOVERY=YES
XTS9-143 CONTROL BASE=S35VMDB1
XTS9-143 RELOAD  CREATOR=SQLDBA,TNAME=PROJECT
XTS9-143         FUNCT=REPLACE
XTS9-143 /*
XTS9-196 Do you want to continue the RELOAD    process ?
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-403 Reply is 1
XTS9-100 Data Restore feature 3.5.0
XTS9-136 Processing S35VMDB1 archived on (05/30/96-15:18:48)
XTS9-182 Following files are needed for recovery
XTS9-195 UARCHIVE     currently mounted
XTS9-179 Current log
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-403 Reply is 1
XTS9-102      1351 rows loaded, procedure completed
XTS9-128        20 rows loaded into (SQLDBA.PROJECT)
    .... more lines ....
XTS9-128        24 rows loaded into (DATARFTR.SYSCOLAUTH)
XTS9-101        10 tables successfully processed
XTS9-314 COMMIT WORK successful for reload of data,
         creating required objects
ALTER TABLE †SQLDBA†.†PROJECT† ADD PRIMARY KEY (†PROJNO† ASC
 ) PCTFREE=10;
COMMIT WORK ;
CREATE INDEX †SQLDBA†.†DEPTNOI† ON †SQLDBA†.†PROJECT†
(†DEPTNO† ASC ) PCTFREE=10;
COMMIT WORK ;
CREATE INDEX †SQLDBA†.†RESPEMPI† ON †SQLDBA†.†PROJECT†
(†RESPEMP† ASC ) PCTFREE=10;
COMMIT WORK ;
ALTER TABLE †SQLDBA†.†PROJ_ACT† ADD FOREIGN KEY †R_PROJ2†
(†PROJNO†) REFERENCES †SQLDBA†.†PROJECT† ON DELETE RESTRICT;
COMMIT WORK ;
ALTER TABLE †SQLDBA†.†PROJECT† ADD FOREIGN KEY †R_DEPT2†
(†DEPTNO†) REFERENCES †SQLDBA†.†DEPARTMENT† ON DELETE RESTRICT;
COMMIT WORK ;
ALTER TABLE †SQLDBA†.†PROJECT† ADD FOREIGN KEY †R_EMPLY2†
(†RESPEMP†) REFERENCES †SQLDBA†.†EMPLOYEE† ON DELETE SET NULL;
COMMIT WORK ;
CONNECT SQLDBA  ;
GRANT SELECT ON †SQLDBA†.†PROJECT† TO †PUBLIC†;
COMMIT WORK ;
CONNECT SQLDBA  ;
COMMIT WORK ;
XTS9-184 Processing current log
XTS9-407 Enter 0(CANCEL),1(CONTINUE) or 111(SKIPFILE)
XTS9-403 Reply is 1
XTS9-007 Processing successfully completed
```

Figure 60. SYSPRINT File of Data Restore RELOAD from Data Restore Archive with Forward Recovery

In Figure 61, ARCHIV is used as a work file while reading the DB2 archive (ARIARCH) from tape. The parameter ARCHTYPE=SQLDS is given in Figure 62.

```
/*-------------------------------------------------------------------*/
/*- Reload procedure without forward recovery ----------------------*/
/*- Input=SQL/DS Archive -------------------------------------------*/
/*-------------------------------------------------------------------*/
¢FILEDEF ARCHIV   DISK ARCHIV DATA A ( RECFM VB BLOCK 32760¢
¢FILEDEF ARIARCH  TAP1 SL            (RECFM FB BLOCK 28672 LRECL 4096¢

¢FILEDEF LMBRLG1 DISK LMBRLG1 DATA A (RECFM VB BLOCK 32760¢
¢FILEDEF LMBRLG2 DISK LMBRLG2 DATA A (RECFM VB BLOCK 32760¢
¢FILEDEF LMBRLG3 DISK LMBRLG3 DATA A (RECFM VB BLOCK 32760¢

¢FILEDEF LMBRWRK DISK LMBRWRK DATA A (RECFM FB BLOCK 28672 LRECL 4096¢
¢FILEDEF SYS0001 DISK SYS0001 DATA A (RECFM F  BLOCK 4096¢
¢FILEDEF HEADER  DISK LMBRWRK DATA A (RECFM F  BLOCK 4096¢
¢FILEDEF DIRWORK DISK LMBRWRK DATA A (RECFM F  BLOCK 4096¢

¢FILEDEF SYSIN    DISK DRFRELA   SYSIN  A¢
¢FILEDEF SYSPRINT DISK DRFRELA   SYSPRINT A¢
¢XTS91001¢
Exit rc
```

Figure 61. EXEC File for Data Restore RELOAD from DB2 Archive without Forward Recovery

```
OPTIONS RECOVERY=NO  CONFIRM=NO ARCHTYPE=SQLDS
CONTROL BASE=S35VMDB1
RELOAD  CREATOR=SQLDBA,TNAME=ACTIVITY,FUNCT=REPLACE
```

Figure 62. SYSIN for Data Restore RELOAD from DB2 Archive without Forward Recovery

```
/**/
¢FILEDEF ARCHIV  TAP1 SL 1 (RECFM VB BLOCK 32760¢

¢FILEDEF SYSIN DISK DESCRIBE SYSIN  A¢
¢FILEDEF SYSPRINT DISK DESCRIBE SYSPRINT A¢
¢XTS91001¢
Exit rc
```

Figure 63. EXEC File for Data Restore DESCRIBE

```
/output
/**/
¢FILEDEF LMBRLG1 DISK LMBRLG1 DATA A (RECFM VB BLOCK 32760¢
¢FILEDEF LMBRLG2 DISK LMBRLG2 DATA A (RECFM VB BLOCK 32760¢
¢FILEDEF LMBRLG3 DISK LMBRLG3 DATA A (RECFM VB BLOCK 32760¢

¢FILEDEF LMBRWRK DISK LMBRWRK DATA T (RECFM FB BLOCK 28672 LRECL 4096¢

¢FILEDEF SYSIN DISK LISTLOG   SYSIN  A¢
¢FILEDEF SYSPRINT DISK LISTLOG   SYSPRINT A¢
¢XTS91001¢
Exit rc
```

Figure 64. EXEC File for Data Restore LISTLOG

```
SELECT * FROM SQLDBA.PROJECT
PROJNO PROJNAME              DEPTNO RESPEMP PRSTAFF
------ -------------------- ------ ------- -------
 .....
IF2000 USER EDUCATION        C01    000030     1.00
 .....
MA2100 WELD LINE AUTOMATION  D01    000010    12.00
 .....
OP1010 OPERATION UPD2        E11    000090     5.00
 .......
OP2010 SYSTEMS SUPPORT UPD1  E21    000100     4.00
 .....
```

Figure 65. Changed Data in Table SQLDBA.PROJECT after Data Restore APPLYLOG

C.4 RELOAD Examples in VSE

Figure 66 shows the JCL used in some of the following examples. It defines the RELOAD work files for forward recovery.

```
* $$ JOB JNM=XTS9DLBL,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB XTS9DLBL  CATALOG DLBL for data restore feature
// EXEC     LIBR,PARM=¢MSHP¢
   ACCESS  SUBLIB=PRD2.RCV350
   CATALOG XTS9DLBL.PROC     EOD=&& DATA=YES REPLACE=YES
// PROC CAT=¢VSESPUC¢
// DLBL IJSYSUC,¢VSESP.USER.CATALOG¢,,VSAM
* $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
* XTS9DLBL: DATA RESTORE FEATURE WORKFILES
* $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
// DLBL LMBRWRK,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG1,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG2,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG3,,,VSAM,CAT=VSESPUC
// DLBL SYS0001,,0,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL HEADER,,0,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL DIRWORK,,0,VSAM,CAT=VSESPUC
&&
/*
/&
* $$ EOJ
```

Figure 66. JCL File DLBL for Data Restore Work Files (XTS9DLBL.PROC)

In Figure 67, the DLBL JCL from Figure 66 is addressed. The Data Restore archive is read from tape. Therefore, SYS006 is assigned and OPTIONS DEVICE=TAPE.

```
* $$ JOB JNM=DRFRLOAD,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFRLOAD              RELOAD TABLE WITH FORWARD RECOVERY
// LIBDEF *,SEARCH=(PRD2.SQL350,PRD2.RCV350)
// EXEC PROC=SQLVSE02        DLBL for database extents
// EXEC PROC=XTS9DLBL        DLBL for drf files
// TLBL     ARCHIV
// ASSGN    SYS006,180       input tape from BACKUP
// MTC      REW,SYS006
// EXEC XTS91001,SIZE=AUTO,PARM=¢DBNAME(SQLVSE02)¢
OPTIONS RECOVERY=YES DEVICE=TAPE
CONTROL DBNAME=SQLVSE02
RELOAD CREATOR=SQLDBA TNAME=PROJECT FUNCT=REPLACE
/*
/&
* $$ EOJ
```

Figure 67. JCL File for Data Restore RELOAD from Data Restore Archive with Forward Recovery

Figure 68 on page 93 shows an example without using the DLBL from above. SYS007 is assigned because a DB2 archive is read. ARCHIV is used as a work file on disk as shown in the DLBL ARCHIV and the OPTIONS DEVICE=DASD statements.

```
* $$ JOB JNM=DRFRLOAD,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFRLOAD                RELOAD TABLE WITH FORWARD RECOVERY
// LIBDEF *,SEARCH=(PRD2.SQL350,PRD2.RCV350)
// EXEC PROC=SQLVSE02
// TLBL     ARIARCH
// ASSGN    SYS007,181             INPUT SQL ARCHIVE
// MTC      REW,SYS007
// DLBL LMBRWRK,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG1,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG2,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG3,,,VSAM,CAT=VSESPUC
// DLBL SYS0001,,,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL HEADER,,,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL DIRWORK,,,VSAM,CAT=VSESPUC
// DLBL ARCHIV,,,VSAM,RECORDS=(100,100),RECSIZE=4096,CAT=VSESPUC
// EXEC XTS91001,SIZE=AUTO,PARM=¢DBNAME(SQLVSE02)¢
OPTIONS RECOVERY=YES,ARCHTYPE=SQLDS,DEVICE=DASD
CONTROL DBNAME=SQLVSE02
RELOAD CREATOR=SQLDBA TNAME=PROJECT FUNCT=REPLACE
/*
/&
* $$ EOJ
```

Figure 68. JCL File for Data Restore RELOAD from DB2 Archive with Forward Recovery

In Figure 69 on page 94 a DB2 archive is read using SYS007 as shown in Figure 68, but the work file ARCHIV is defined on tape. Refer to the TLBL ARCHIV and ASSGN SYS006 statements.

```
* $$ JOB JNM=SQLRLOAD,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB SQLRLOAD    RELOAD 2 TABLEs WITH FORWARD REC from sql archi
// LIBDEF *,SEARCH=(PRD2.SQL350,PRD2.RCV350)
// EXEC PROC=SQLVSE02
// EXEC PROC=XTS9DLBL
// TLBL     ARCHIV
// TLBL     ARIARCH
// TLBL     LARCHIV
// ASSGN    SYS006,180
// ASSGN    SYS007,181
// MTC      REW,SYS007
// EXEC XTS91001,SIZE=AUTO,PARM=¢DBNAME(SQLVSE02)¢
OPTIONS ARCHTYPE=SQLDS RECOVERY=YES
CONTROL DBNAME=SQLVSE02
RELOAD CREATOR=SQLDBA TNAME=PROJECT FUNCT=REPLACE
RELOAD CREATOR=SQLDBA TNAME=PROJ_ACT FUNCT=REPLACE
/*
/&
* $$ EOJ
```

Figure 69. JCL File for Data Restore RELOAD of 2 Tables from DB2 Archive with Forward Recovery

The guest sharing example shows the general use of the PARM='DBNAME()" statement as opposed to CONTROL DBNAME=. PARM indicates the target database, whereas the CONTROL statement defines the source of the archive, that is, the database from which the archive was taken.

```
* $$ JOB JNM=DRFRLOAD,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFRLOAD    RELOAD TABLE with guest sharing
// LIBDEF *,SEARCH=(PRD2.SQL350,PRD2.RCV350)
// TLBL     ARCHIV                * input tape *
// ASSGN    SYS006,181
// MTC      REW,SYS006
// SETPFIX  LIMIT=1M
// EXEC XTS91001,SIZE=AUTO,PARM=¢DBNAME(S35VMDB1)¢
OPTIONS RECOVERY=NO,DEVICE=TAPE
CONTROL DBNAME=SQLVSE02
RELOAD CREATOR=SQLDBA TNAME=ACTIVITY FUNCT=REPLACE
/*
/&
* $$ EOJ
```

Figure 70. JCL File for Data Restore RELOAD with Guest Sharing

C.5 RESTORE Examples in VSE

```
* $$ JOB JNM=DRFRESTR,CLASS=7,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,ELADM1)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,ELADM1)
// JOB DRFRESTR               RESTORE FULL DATABASE from tape
// LIBDEF *,SEARCH=(PRD2.DB2510,PRD2.RCV510)
// EXEC PROC=SQLVSE02
// EXEC PROC=XTS9DLBL
// TLBL     ARCHIV
// ASSGN    SYS006,181
// MTC      REW,SYS006
/* PAUSE for monitor
// EXEC XTS91001,SIZE=AUTO
CONTROL DBNAME=SQLVSE02
RESTORE
/*
/&
* $$ EOJ
```

Figure 71. JCL File for Data Restore RESTORE of a Full Database

```
* $$ JOB JNM=DRFRESTR,CLASS=7,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,ELADM1)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,ELADM1)
// JOB DRFRESTR               RESTORE STORAGE POOL from BACKUP tape
// LIBDEF *,SEARCH=(PRD2.DB2510,PRD2.RCV510)
// EXEC PROC=SQLVSE02
// EXEC PROC=XTS9DLBL
// TLBL     ARCHIV
// ASSGN    SYS006,181
// MTC      REW,SYS006
/* PAUSE for monitor
// EXEC XTS91001,SIZE=AUTO
CONTROL DBNAME=SQLVSE02
RESTORE POOL=8
/*
/&
* $$ EOJ
```

Figure 72. JCL File for Data Restore RESTORE of a Storage Pool

```
// JOB DRFRESTR               DRF 4.1 RESTORE STORAGE POOL FROM BACKUP
// LIBDEF *,SEARCH=(PRD2.SQL410,PRD2.RCV410)
// EXEC PROC=SQLVSE02
// PROC CAT=¢SQLPCAT¢
// DLBL IJSYSUC,¢SQLPCAT.USER.CATALOG¢,,VSAM
* ******************************************************
* SQLVSE02: SQL/DS DATABASE IDENTIFICATION
* ******************************************************
// DLBL BDISK,¢SQLVSE02.BDISK.SQLDIR80¢,,VSAM,CAT=SQLPCAT
// DLBL LOGDSK1,¢SQLVSE02.LOGDSK1.SQLLOG¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK1,¢SQLVSE02.DDSK1.POOL1¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK2,¢SQLVSE02.DDSK2.POOL2¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK3,¢SQLVSE02.DDSK3.POOL3¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK4,¢SQLVSE02.DDSK4.POOL4¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK5,¢SQLVSE02.DDSK5.POOL5¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK6,¢SQLVSE02.DDSK6.POOL6¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK7,¢SQLVSE02.DDSK7.POOL7¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK8,¢SQLVSE02.DDSK8.POOL8¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK9,¢SQLVSE02.DDSK9.POOL9¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK10,¢SQLVSE02.DDSK10.POOL10¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK11,¢SQLVSE02.DDSK11.POOL11¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK12,¢SQLVSE02.DDSK12.POOL12¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK13,¢SQLVSE02.DDSK13.POOL13¢,,VSAM,CAT=SQLPCAT
// DLBL DDSK14,¢SQLVSE02.DDSK14.POOL14¢,,VSAM,CAT=SQLPCAT
EOP SQLVSE02
// EXEC PROC=XTS9DLBL
// PROC CAT=¢VSESPUC¢
// DLBL IJSYSUC,¢VSESP.USER.CATALOG¢,,VSAM
* ******************************************************
* XTS9DLBL: DATARESTORE FEATURE WORKFILES
* ******************************************************
// DLBL LMBRWRK,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG1,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG2,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG3,,,VSAM,CAT=VSESPUC
// DLBL SYS0001,,0,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL HEADER,,0,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL DIRWORK,,0,VSAM,CAT=VSESPUC
EOP XTS9DLBL
// TLBL     ARCHIV
// ASSGN    SYS006,181
// MTC      REW,SYS006
// EXEC XTS91001,SIZE=AUTO

XTS9-143 CONTROL DBNAME=SQLVSE02
XTS9-143 RESTORE POOL=8
XTS9-143 /*
XTS9-100 Data restore feature version 4.1.0
XTS9-136 Processing SQLVSE02 archived on (02/19/97-15:53:07)
XTS9-182 Following files are needed for recovery
XTS9-195 UARCHIVE     currently mounted
XTS9-179 Current log
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-403 Reply is 1
XTS9-211 Beginning update of directory
XTS9-006 Processing DDSK8
XTS9-010      2452 blocks restored
XTS9-307 Start the database manager with parameter †STARTUP=U†
XTS9-007 Processing successfully completed
1S55I  LAST RETURN CODE WAS 0000
EOJ DRFRESTR  MAX.RETURN CODE=0000
```

Figure 73. Output File from Data Restore RESTORE of a Storage Pool

C.6 RESTORE Example in VM

```
/* */
/* FILEDEF FOR INPUT FROM TAPE */
¢FILEDEF ARCHIV   TAP1 SL (RECFM VB BLOCK 32760¢
¢FILEDEF SYSIN    DISK RESPOOL SYSIN A¢
¢FILEDEF DIRWORK  DISK DIRWORK DATA  A¢
¢FILEDEF SYSPRINT DISK RESPOOL SYSPRINT A¢
¢XTS91001¢
exit rc
```

Figure 74. EXEC File for Data Restore RESTORE of a Storage Pool from Tape (RESPOOL EXEC)

```
OPTIONS RECOVERY=NO  CONFIRM=NO
CONTROL DBNAME=ELDB2A
RESTORE POOL=8
```

Figure 75. SYSIN File for Data Restore RESTORE of a Storage Pool from Tape (RESPOOL SYSIN)

C.7 Example of Data Restore TRANSLATE Using Control Center

The steps which follow show how to TRANSLATE a database archive into a Data Restore BACKUP format using Control Center.

To invoke a TRANSLATE, select the Database Utilities Menu (**U**), followed by **DR**. Or, using fastpath, enter **U.DR**, from the Control Center Main Menu (see Figure 76 on page 98).

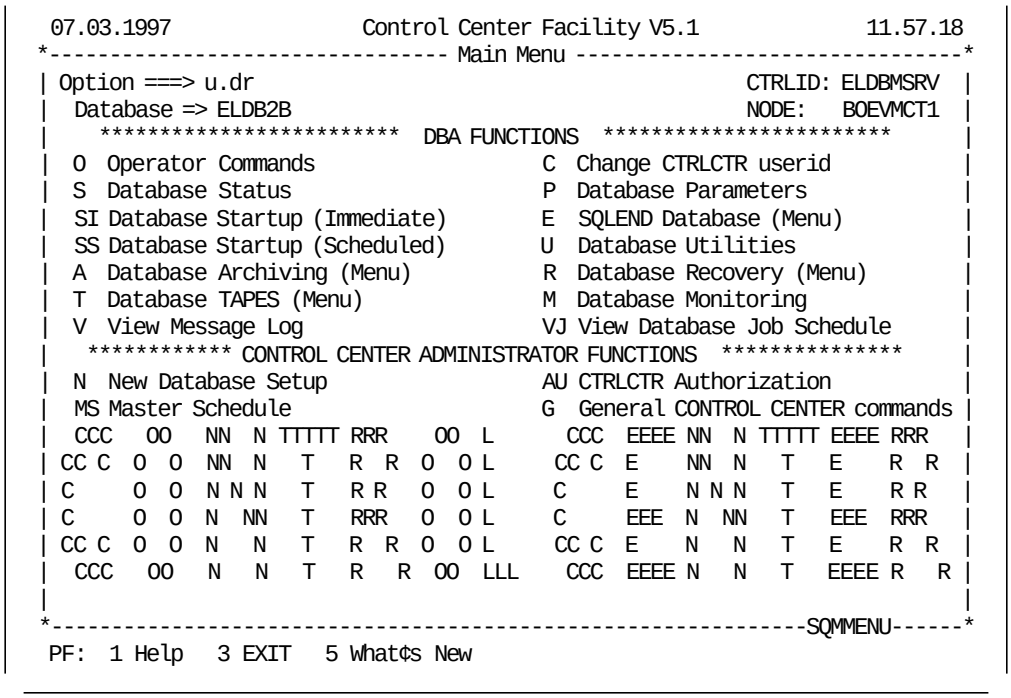


Figure 76. Invoking Data Restore

Before invoking TRANSLATE, you must first choose the archive series you wish to TRANSLATE. You must also define your translate output (disk or tape) and control files. This can be accomplished by editing the Database Tapes File.

To begin this step, type **TM** from the Data Restore menu as illustrated in Figure 77 on page 99.

```
 07.03.1997                Control Center Facility                11.58.27
*------------------------------ Data Restore Menu -------------------------*
| Option ===> tm                                          CTRLID: ELDBMSRV |
|    Database ===> ELDB2B                                 NODE:   BOEVMCT1 |
|                                                         DRMACH: ELDB2DRF |
|                                                                          |
|     T  TRANSLATE ARCHIVE          TRANSLATE archive into BACKUP format   |
|     U  UNLOAD DBSPACES            UNLOAD one or more dbspaces            |
|     R  RELOAD TABLES              RELOAD one or more tables              |
|     LL LISTLOG                    LISTLOG selection panel                |
|     AL APPLYLOG                   APPLYLOG selection panel               |
|     VJ VIEW JOB SCHEDULE          VIEW database job schedule             |
|     S  VIEW DRMACH STATUS         VIEW Data Restore Machine status       |
|     SR RESET DRMACH STATUS        RESET Data Restore Machine status      |
|                                                                          |
|          View Tapes   Edit Tapes    View History    View Log            |
|  BACKUP      BT          BM             BH             BL               |
|  UNLOAD      UT          UM             UH             UL               |
|  TRANSLATE   TT          TM             TH             TL               |
|                                                                          |
|          Enter OPTION, select DATABASE, press ENTER to process           |
|                                                                          |
*--------------------------------------------------------------SDRMAIN-----*
 PF:   1 Help    3 End
```

Figure 77. Choosing Archive Series and Defining TRANSLATE Tapes

From the Database TAPES File, we have chosen to TRANSLATE the archive entry (VOL100) from series 100. Next, we have added two entries for series 100 to handle the TRANSLATE process. The first entry, TRANS, must be specified for the TRANSLATE output file (either disk or tape). In our case, we have selected tape (Volid TRL200). The second entry must be disk (TRANSDSK); the filename, filetype and cuu must be provided (ELRES2 TRANSLAT 192). This file is where the Data Restore control information for SYS0001, HEADER, and DIRWORK, will be found.

Note: The series number must be the same for each archive and its associated output and control files used for the translation.

```
 07.03.1997                Control Center Facility                 11.59.45
*---------------------- Database  TAPES   File Update ----------------------*
| Command ===>                                                CTRLID: ELDBMSRV |
|  Database => ELDB2B                                         NODE:   BOEVMCT1 |
|                                        VOLID/                               |
| SERIES  TYPE     DATE   TIME     STATUS   FILENAME  FILETYPE  FM  SCRCUU    |
|  100    ARCHIVE  97066  11:50:14 FILDEF   VOL100                            |
|  100    LOG      00000  00:00:00 UNUSED   LOG100                            |
|  100    LOG      00000  00:00:00 UNUSED   LOG101                            |
|  100    TRANS    00000  00:00:00 UNUSED   TRL200                            |
|  100    TRANSDSK 00000  00:00:00 UNUSED   ELRES2    TRANSLAT  *    192     |
|  200    ARCHIVE  97066  10:54:47 FILLED   VOL200                            |
|  200    LOG      97066  10:51:13 FILLED   LOG200                            |
|  200    LOG      00000  00:00:00 UNUSED   LOG201                            |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|   Make changes, place D in SERIES to DELETE , press PF10 to process         |
|                           Page 1     of 1                                   |
|                                                                             |
*-----------------------------------------------------------------SQMTP20-----*
 PF:  1 Help   3 End    4 Add Tape   7 Bkwd     8 Fwd     10 Process updates
```

Figure 78. Database Tapes File

After updating the Database Tapes File, to start the TRANSLATE function select **T** from the Data Restore Menu.

```
 07.03.1997                Control Center Facility                 12.01.15
*------------------------------ Data Restore Menu -------------------------*
| Option ===> t                                               CTRLID: ELDBMSRV |
|    Database ===> ELDB2B                                     NODE:   BOEVMCT1 |
|                                                             DRMACH: ELDB2DRF |
|                                                                              |
|     T  TRANSLATE ARCHIVE          TRANSLATE archive into BACKUP format       |
|     U  UNLOAD DBSPACES            UNLOAD one or more dbspaces                |
|     R  RELOAD TABLES              RELOAD one or more tables                  |
|     LL LISTLOG                    LISTLOG selection panel                    |
|     AL APPLYLOG                   APPLYLOG selection panel                   |
|     VJ VIEW JOB SCHEDULE          VIEW database job schedule                 |
|     S  VIEW DRMACH STATUS         VIEW Data Restore Machine status           |
|     SR RESET DRMACH STATUS        RESET Data Restore Machine status          |
|                                                                              |
|          View Tapes   Edit Tapes    View History    View Log                |
|  BACKUP      BT          BM             BH             BL                    |
|  UNLOAD      UT          UM             UH             UL                    |
|  TRANSLATE   TT          TM             TH             TL                    |
|                                                                              |
|           Enter OPTION, select DATABASE, press ENTER to process              |
|                                                                              |
*-----------------------------------------------------------------SDRMAIN-----*
 PF:   1 Help    3 End
```

Now the archive set to be translated needs to be specified. To view all the Archive Sets, type **ALL**, or press enter to see the current Archive Sets from the menu shown in Figure 79 on page 101.

```
 07.03.1997              Control Center Facility                12.02.46
*---------------------- Select Archive for Translation --------------------*
| Command ==>                                              CTRLID: ELDBMSRV |
|    Database ===> ELDB2B                                  NODE:   BOEVMCT1 |
|                                                          DRMACH  ELDB2DRF |
|                                                                           |
|    Translate Set ===>          Archive Set for Translation                |
|                              ( Blank for LATEST or ALL for all available ) |
|         To TRANSLATE a database archive, you must view the Restore        |
|    Set Report and select the archive set for translation.  The translate  |
|    set will consist of tapes or disks from a previous database archive.   |
|    You may choose ALL to display all for all available archive sets,      |
|    or you may leave the translate set BLANK to retrieve the LATEST        |
|    archive.                                                               |
|                                                                           |
|    The selected archive will be displayed, and you will be asked to       |
|    enter the archive set number to begin the TRANSLATE.                   |
|                                                                           |
|        Enter Translate Set, press ENTER to process, or                    |
|        press PF3 to QUIT                                                  |
|                                                                           |
*---------------------------------------------------------------SQMTR60-----*
 PF:   1 Help    3 End (Quit)
```

Figure 79. Select Archive Set for Translation

From this menu (Figure 79), you will be placed in PEEK mode to review the Restore Set Report. This report will show the date, timestamp, series number, and the volid(s) belonging to each archive restore set.

```
RESTORE SET REPORT for Data Base ELDB2B      Date: 07.03.1997  Time: 15.36.43

RESTORE SET(s) generated using LOGMODE = L

  NOTE:  Examine the following restore set(s) and
       remember the restore set that you wish to use.


         Restore Set #1   From Data Base Archive Created 07.03.1997 15.03.11 ...

            Archive          Archive        Date      Time
        Type (DB or Log)    Sequence     Archived  Archived Series Volid
        ________________  ________  __________ ________ ______ ______
        Data Base Archive  Tape #1   07.03.1997 15.03.11   100  VOL100
                                  ..................................
        ACTIVE Log         >>>>    Archived During The Recovery    <<<<

         Restore Set #1 END ..............................................
```

The Translate Set Number specified must match the appropriate Series in the Database Tapes File. In our case, Translate Set 1 is associated with series 100, as both the Restore Set Report and Database Tapes file reflect.

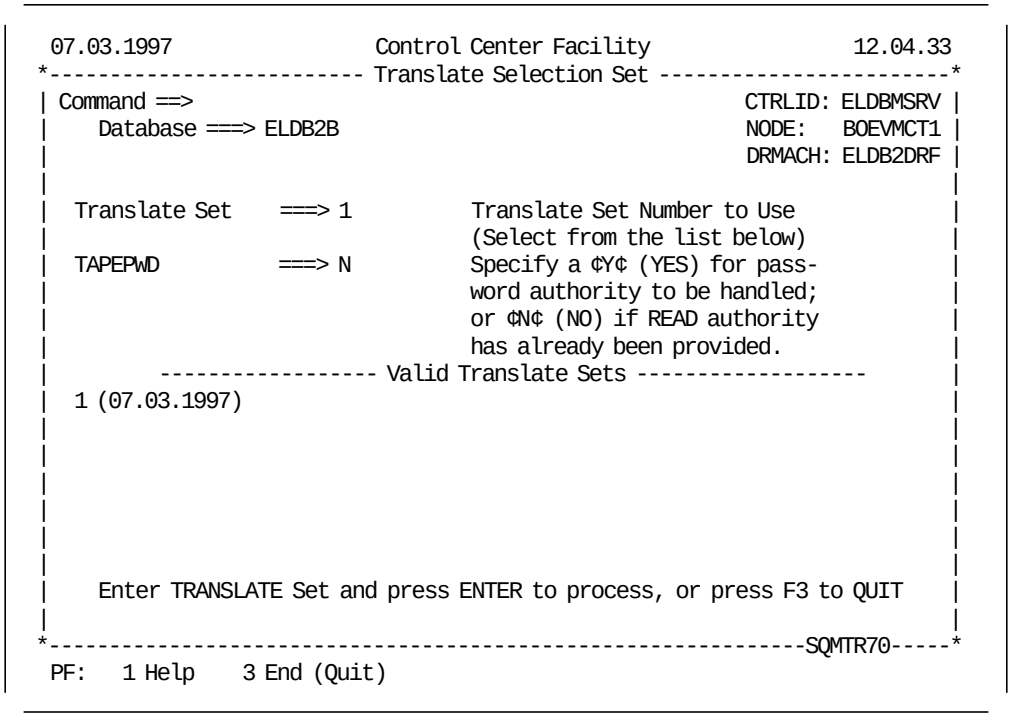
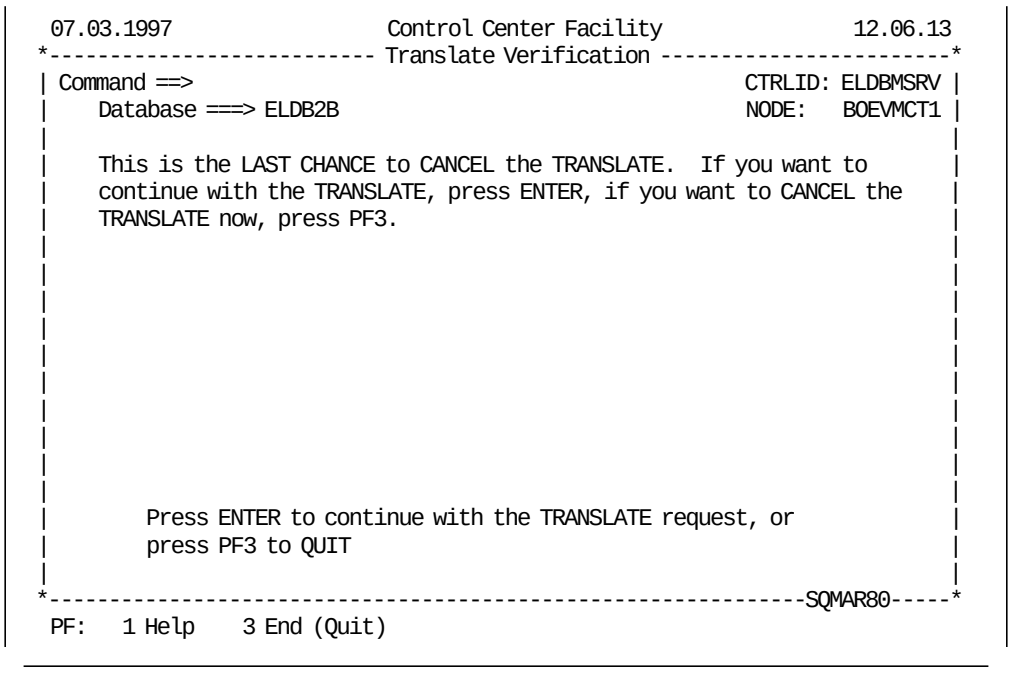


Figure 80. Translate Selection Set Menu

Before submitting the TRANSLATE job, there is one last chance to cancel or quit.



After the TRANSLATE job has been submitted, its log will be available for review. To browse this log, select **TL** from the Data Restore Menu, shown in Figure 77 on page 99.

The following Translate Log, shows the job we submitted. The TRANSLATE file created can now be used as input for either a RESTORE or RELOAD function with Control Center.

```
XTS9-143 * 03/07/1997              CONTROL CENTER                  15:37:43
XTS9-143 *                        TRANSLATE ARCHIVE
XTS9-143 * SYSIN FILE: ELDB2B TRANSYIN A
XTS9-143 * :DBMACH= ELDB2B DRMACH= ELDB2DRF CTLMACH= SQLMSTR
XTS9-143 * ARCHIVE: TAPE SERIES 100 FROM 07.03.1997 15.03.11
XTS9-143 * TRANS:  TRANMEDIA
XTS9-143 * TRANSDSK: ELRES2 SYS001/HEADER/DIRWORK G 192 121
XTS9-143 *
XTS9-143 * SYSPRINT DISK     ELDB2B   TRANLIST A1
XTS9-143 * SYSIN    DISK     ELDB2B   TRANSYIN A1
XTS9-143 * ARCHIV   TAP5  SL  00001  VOLID TRL200
XTS9-143 * Z        DISK     DMSNAM   LOADLIB  *
XTS9-143 * SYS0001  DISK     ELRES2   SYS0001  G1
XTS9-143 * HEADER   DISK     ELRES2   HEADER   G1
XTS9-143 * DIRWORK  DISK     ELRES2   DIRWORK  G1
XTS9-143 * ARIARCH  TAP1  SL  00001  VOLID VOL100
XTS9-143 *
XTS9-143 * DDNAME   VOLID   FSEQ VOLSEQ GENN GENV  CRDTE  EXDTE SEC    FID
XTS9-143 * ARCHIV   TRL200
XTS9-143 * ARIARCH  VOL100
XTS9-143 * ********************************************
XTS9-143 OPTIONS CONFIRM=YES NOTATION=E LANG=S001
XTS9-143       MSGCLASS=1 MSGDEV=3 CASE=M
XTS9-143 CONTROL DBNAME=ELDB2B
XTS9-143 TRANSLATE
XTS9-143 /*
XTS9-196 Do you want to continue the TRANSLATE process ?
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-403 Reply is 1
XTS9-100 Data restore feature version 4.1.0
XTS9-193 Mount first tape of database server archive
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-403 Reply is 1
XTS9-013 Table SQLDBA  .ACTIVITY          may be reloaded
XTS9-013 Table DATARFTR.CMD               may be reloaded
XTS9-013 Table SQLDBA  .DEPARTMENT        may be reloaded
XTS9-013 Table SQLDBA  .DRIVER_TABLE      may be reloaded
XTS9-013 Table SQLDBA  .ELOLANGUAGE       may be reloaded
XTS9-013 Table SQLDBA  .ELOOPTIONS        may be reloaded
XTS9-013 Table SQLDBA  .ELOTEXT1          may be reloaded
XTS9-013 Table SQLDBA  .ELOTEXT2          may be reloaded
XTS9-013 Table SQLDBA  .EMP_ACT           may be reloaded
XTS9-013 Table SQLDBA  .EMPLOYEE          may be reloaded
XTS9-013 Table SQLDBA  .INVENTORY         may be reloaded
XTS9-013 Table SQLDBA  .ITEM              may be reloaded
XTS9-013 Table SQLDBA  .ITEM1             may be reloaded
XTS9-013 Table SQLDBA  .IXLNGVAR          may be reloaded
XTS9-013 Table SQLDBA  .LMBRINS           may be reloaded
```

```
XTS9-013 Table SQLDBA  .OPERATIONS        may be reloaded
XTS9-013 Table SQLDBA  .PROJ_ACT          may be reloaded
XTS9-013 Table SQLDBA  .PROJECT           may be reloaded
XTS9-013 Table SQLDBA  .PROJECTS          may be reloaded
XTS9-013 Table SQLDBA  .QUOTATIONS        may be reloaded
XTS9-013 Table EXAMPLE .ROUTINE           may be reloaded
XTS9-013 Table SQLDBA  .ROUTINE           may be reloaded
XTS9-013 Table SQLDBA  .STORED QUERIES    may be reloaded
XTS9-013 Table SQLDBA  .SUPPLIERS         may be reloaded
XTS9-013 Table DATARFTR.SYSCATALOG        may be reloaded
XTS9-013 Table DATARFTR.SYSCOLAUTH        may be reloaded
XTS9-013 Table DATARFTR.SYSCOLUMNS        may be reloaded
XTS9-013 Table DATARFTR.SYSINDEXES        may be reloaded
XTS9-013 Table DATARFTR.SYSKEYCOLS        may be reloaded
XTS9-013 Table DATARFTR.SYSKEYS           may be reloaded
XTS9-013 Table SQLDBA  .SYSLANGUAGE       may be reloaded
XTS9-013 Table DATARFTR.SYSTABAUTH        may be reloaded
XTS9-013 Table SQLDBA  .SYSTEXT1          may be reloaded
XTS9-013 Table SQLDBA  .SYSTEXT2          may be reloaded
XTS9-013 Table DATARFTR.SYSUSAGE          may be reloaded
XTS9-013 Table DATARFTR.SYSVIEWS          may be reloaded
XTS9-007 Processing successfully completed
```

Figure 81 (Part 2 of 2). View Translate Log Using Control Center

For more information about the TRANSLATE operation, as well as the other Data Restore functions using Control Center, please refer to the appropriate chapters of the *Control Center for VM Operations and Installation Guide*.

C.8 Example of Data Restore RELOAD Using Control Center

RELOAD allows the user to reload and recreate tables from a Data Restore BACKUP, translated database archive, or a Data Restore UNLOAD. Control Center requires that a database archive first be translated before it is used as an input source for RELOAD.

In our example the RECOVERY option is set to ′YES′. Then Data Restore will read the active log to record all changes to this table executed since the BACKUP into work files. This can later be examined with the LISTLOG function and re-executed using the APPLYLOG function.

To run the Data Restore RELOAD the database must be active.

In this example we used a translated archive tape as input to the RELOAD process.

To start the RELOAD function, select **R** from the Date Restore Menu as Figure 82 on page 105 shows.

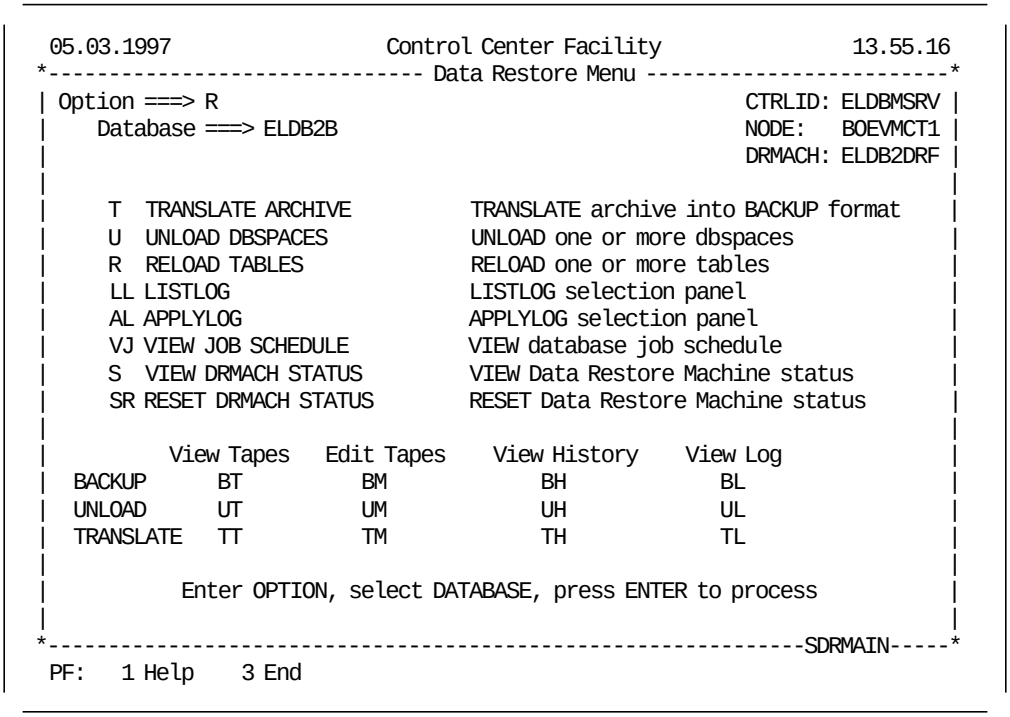


Figure 82. Invoking Data Restore RELOAD Function

On the screen bellow, we specified Log Recovery=1 (Yes) and the input type is a translated archive tape.

```
 05.03.1997             Control Center Facility                    14.02.19
*------------------  Data Restore Reload Table Utility  ---------------------*
|                                                          CTRLID:  ELDBMSRV |
| Database ==> ELDB2B                                      NODE:    BOEVMCT1 |
|                                                          DRMACH:  ELDB2DRF |
|                                                                            |
|                                                                            |
|    Log Recovery ==> 1         ( 1 = ¢YES¢, 2 = ¢NO¢                        |
|    Input Type   ==> 2         ( 1 = BACKUP, 2 = TRANSLATE, 3 = UNLOAD      |
|    Restart      ==> 2         ( 1 = ¢YES¢, 2 = ¢NO¢, Restart RELOAD        |
|                                     and use the same RELDCTL file          |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
*----------------------------------------------------------------DRRELD2------*
  PF1 HELP    PF3 QUIT    PF4 EXIT    PF5 Main Menu    Enter Retrieve List
```

To see all available restore sets, type **all** on the RELOAD SET field. A restore set report will be presented. Write down the restore set number you wish to use. In our case, the restore set number **1** will be used.

```
 05.03.1997               Control Center Facility            14.06.06
*----------------------- Reload Restore Set Selection -----------------------*
| Command ==>                                                SQMID: ELDBMSRV  |
|    Database ===> ELDB2B                                    NODE:  BOEVMCT1  |
|                                                                             |
|                                                                             |
|    RELOAD SET ===> all               Archive Restore Set to RELOAD          |
|                  ( Blank for LATEST or ALL for all available )              |
|         To execute RELOAD, you must select the archive restore set          |
|    to use.  The restore set is a group of tapes or disks that were used     |
|    for a previous BACKUP or TRANSLATE.  You may choose ALL to display ALL   |
|    available restore sets, or you may leave the restore set field BLANK     |
|    to retrieve the LATEST restore set.                                      |
|                                                                             |
|    The selected restore set will be displayed and you will be asked to      |
|    enter the restore set number to begin the RELOAD.                        |
|                                                                             |
|        Enter Restore Set number and press ENTER to process, or press        |
|         PF3 to QUIT                                                         |
|                                                                             |
*-----------------------------------------------------------------SQMAR60------*
 PF:   1 Help    3 End (Quit)
```

```
RESTORE SET REPORT for Data Base ELDB2B    Date: 07.03.1997  Time: 18.49.05
DRF  1 2 3

RESTORE SET(s) generated using LOGMODE = L

  NOTE:  Examine the following restore set(s) and
       remember the restore set that you wish to use.


         Restore Set #1   From Data Base Archive Created 07.03.1997 15.03.11

            Archive          Archive       Date      Time
        Type (DB or Log)    Sequence    Archived  Archived Series Volid
        _________________  _________  __________ ________ ______ ______
        Data Base Archive  Tape #1   07.03.1997 15.03.11   100  VOL100
                           TDSK @1   07.03.1997 15.39.35   100  ELRES2
TRANSLAT * 192
                           BAC1 @2   07.03.1997 15.41.01   100  TRL200
                                ..................................
        ACTIVE Log         >>>>    Archived During The Recovery    <<<<

         Restore Set #1 END ..........................................

         Restore Set #2   From Data Base Archive Created 07.03.1997 10.54.48

            Archive          Archive       Date      Time
        Type (DB or Log)    Sequence    Archived  Archived Series Volid
        _________________  _________  __________ ________ ______ ______
        DB USER ARCHIVE    Uarc #1   07.03.1997 10.54.48   200  USER BACKUP
DRFBACKUP 03/07/97 10:51:13

        Log Archive(s):    Log  #2   07.03.1997 15.00.19   200  LOG201
                                ..................................
        ACTIVE Log         >>>>    Archived During The Recovery    <<<<

         Restore Set #2 END ..........................................

         Restore Set #3   From Data Base Archive Created 07.03.1997 10.19.58

            Archive          Archive       Date      Time
        Type (DB or Log)    Sequence    Archived  Archived Series Volid
        _________________  _________  __________ ________ ______ ______
        DB USER ARCHIVE    Uarc #1   07.03.1997 10.19.58   200  USER BACKUP
DRFBACKUP 03/07/97 10:16:23
                           BAC1 @1   07.03.1997 10.19.56   200  VOL002
                                ..................................
        ACTIVE Log         >>>>    Archived During The Recovery

         Restore Set #3 END ..........................................
```

Figure 83. Restore Set Report

On the RELOAD Restore Set Selection screen, we type the restore set number **1** to be used. In our case, the database is running in logmode=L and we do not have any log tapes. Just the active log will be processed. If you have 1 or more log tapes, you can limit the number of log tapes to be processed by especifing **Y** on the Process Partial Logs field.

```
 05.03.1997               Control Center Facility            14.08.12
*---------------------- RELOAD Restore Set Selection ----------------------*
| Command ==>                                                SQMID: ELDBMSRV  |
|    Database ===> ELDB2B                                    NODE:  BOEVMCT1  |
|                                                                             |
|  Logmode          ===> L            Logmode for the RELOAD                  |
|  RELOAD Set       ===> 1            Archive Restore Set Number to Use       |
|                                     (Select from the list below)            |
|  Process Partial Logs   ===> N      Limit Logs Processed (Y or N)           |
|                                                                             |
|  BACKUP Type       ===> BACKUP      BACKUP or BACKUP2                       |
|        ------------------ Valid Restore Sets -------------------            |
|  1 (04.03.1997), 2 (04.03.1997), 3 (04.03.1997)                             |
|                                                                             |
|                                                                             |
|   NOTE:   If RECOVERY=¢YES¢, Data Restore will process the logs.  To        |
|           limit the number of logs read, ENTER Y for Partial Log            |
|           processing.                                                       |
|                                                                             |
|    Enter Restore Set and press ENTER to process, or press PF3 to QUIT       |
|                                                                             |
*-----------------------------------------------------------------SQMAR70------*
 PF:   1 Help    3 End (Quit)
```

Press **enter** to continue the RELOAD process.

```
 05.03.1997               Control Center Facility                14.13.08
*-------------------------- Reload Verification ---------------------------*
| Command ==>                                                SQMID: ELDBMSRV  |
|    Database ===> ELDB2B                                    NODE:  BOEVMCT1  |
|                                                                             |
|    This is your chance to review the RELOAD set.  If you want to            |
|    continue with the RELOAD, press ENTER, if you want to CANCEL the         |
|    RELOAD now, press PF3.                                                   |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|        Press ENTER to continue with the RELOAD request, or                  |
|        press PF3 to QUIT                                                    |
|                                                                             |
*-----------------------------------------------------------------SQMAR80------*
 PF:   1 Help    3 End (Quit)
```

The next panel is the RELOAD Table List screen. Here, you specify the table(s) to be reloaded. Each table must include the function (Purge, New, Add, Replace) to be applied. In this example, the sample sqldba.activity table will be reloaded. You can specify up to 90 RELOAD table statements. After entering the RELOAD table commands, press **PF5** to continue the execution of the RELOAD process.

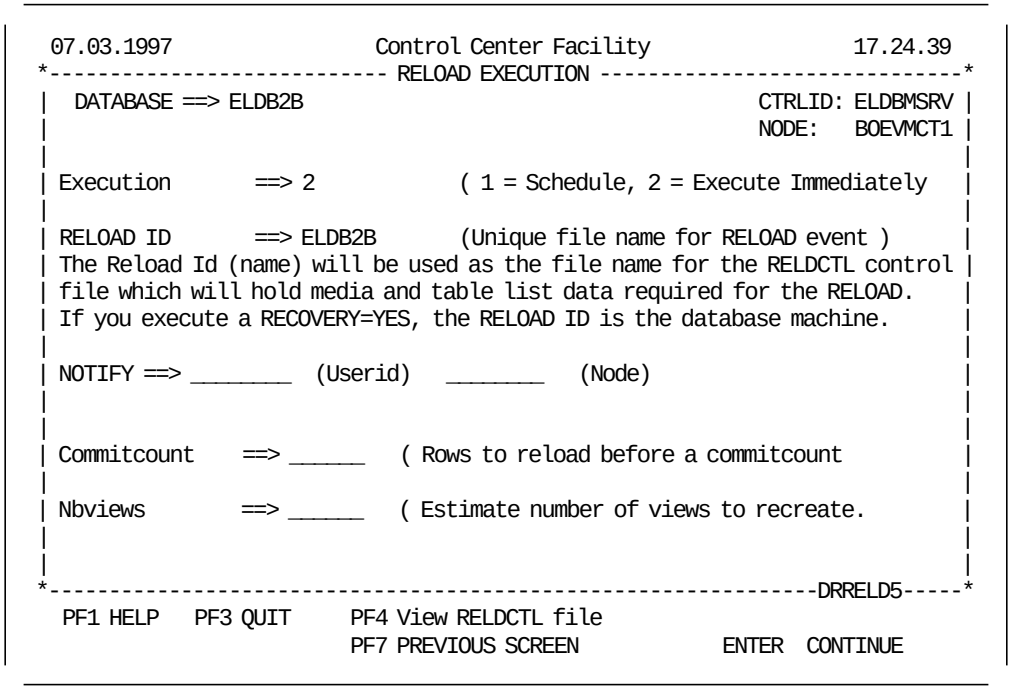
```
 05.03.1997                Control Center Facility                  14.15.21
*---------------------------- Reload Table List -----------------------------*
|  Database => ELDB2B                                Selected Tables => 0    |
|     P,N,A,R   Table                              Dbspace                   |
| Num  Funct    Creator   Table Name          Owner      Dbspace Name        |
| ---  -----    --------  ------------------  --------   ------------------  |
|       P       sqldba__  activity_________  public__   sample____________  |
|          NEW  ________  ____________________                               |
|                                                                            |
|       _       ________  ____________________  ________   ________________  |
|          NEW  ________  ____________________                               |
|                                                                            |
|       _       ________  ____________________  ________   ________________  |
|          NEW  ________  ____________________                               |
|                                                                            |
|       _       ________  ____________________  ________   ________________  |
|          NEW  ________  ____________________                               |
|                                                                            |
|                                                                            |
|                             Page 1     of 1                                |
*----------------------------------------------------------------DRRELD3------*
PF:  1 Help         3 QUIT     5 Continue     6 Previous Menu
```

In the next step, Control Center will show the RELOAD control file that is created to perform the RELOAD function. Here we can check if the right files (tape and disk) are being used. Press **PF3** to exit from the file and continue with the next panel.

```
 $$RELOAD $RELDCTL A1  V 74             1 Blks  3/07/97 Line       1 of 5
===>
:DBMACH=ELDB2B RELOAD RECOVERY=YES from TRANSLATEREC @ 07.03.1997 15.03.11
#1 DRF TRANSDSK 07.03.1997 15.39.35 100 N  ELRES2 TRANSLAT * 192
#2 DRF BACKUP 07.03.1997 15.41.01 100 N  TRL200
#3 LOG Archive ACTIVE LOG
END RESTORE SET
* * * End of file * * *
```

Figure 84. Reload Control File

Below we have the last panel of the RELOAD function. On this panel you specify the type of execution. In our case we will execute the RELOAD immediately. Also, In this example we are using RECOVERY=YES, therefore the RELOAD ID is name of the database machine (ELDB2B) and you cannot change it. For RECOVERY=NO, this field is empty and you can enter any unique file name.



After Control Center finishes the RELOAD operation, the LISTLOG function can be used to produce a report of the changes to the tables that were specified at the RELOAD operation. In our case, the sample sqldba.activity table was used. The report produced with the LISTLOG function can be used to decide which LUWs will be re-applied to the database using the APPLYLOG function.

Appendix D. Special Notices

This publication is intended to help system programmers and SQL/DS database administrators to use the Data Restore feature and to design their archiving strategy.

The information in this publication is not intended as a specification of any programming interfaces that are provided by DB2 Server for VSE & VM V5R1 and the Data Restore feature. See the PUBLICATIONS section of the IBM Programming Announcement for DB2 V3R5 and Data Restore feature for more information about what publications are considered to be product documentation.

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, 500 Columbus Avenue, Thornwood, NY 10594 USA.

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 information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. 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 performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

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

APPN	DB2
DFSMS	IBM
ILE	Nways
QMF	SKI
SQL/DS	System/390
VM/ESA	VSE/ESA
VTAM	

The following terms are trademarks of other companies:

C-bus	Corollary, Inc.
PC Direct	Ziff Communications Company (used by IBM Corporation under license)
UNIX	X/Open Company Ltd. (registered trademark in the United States and other countries)
Windows, Windows 95 logo	Microsoft Corporation

Appendix E. Related Publications

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

E.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 115.

- *VTAM 4.2 Implementation and Usage for VM/ESA and VSE/ESA*, SG24-4556
- *3746 Nways Controller M 950 and M 900 APPN Implementation Guide*, SG24-2536
- *3174 APPN Implementation Guide Update*, SG24-4171
- *SQL/DS Archive and Recovery Using the Data Restore Feature*, SG24-4833.

E.2 Other Publications

These publications are also relevant as further information sources.

- *SQL/DS Data Restore Guide Version 3 Release 5*, SC09-2275
- *SQL/DS System Administration for VSE*, GH09-8096
- *SQL/DS System Administration for IBM VM Systems*, GH09-8084
- *SQL/DS Database Administration for VSE*, GH09-8095
- *SQL/DS Database Administration for IBM VM Systems*, GH09-8083
- *SQL/DS Database Service Utility for IBM VM Systems*, SH09-8088
- *SQL/DS Database Service Utility for VSE*, SH09-8100
- *SQL/DS Diagnosis Guide and Reference for IBM VM Systems*, LH09-8081
- *SQL/DS Diagnosis Guide and Reference for VSE*, LH09-8093
- *SQL Master Installation and Operations Guide for VSE*, SC17-2000-00
- *SQL Master Installation and Operations Guide for IBM VM Systems*, SC67-0205
- *VSE System Control Statements*, SC33-6513
- *VSE/Virtual Storage Access Method: Commands and Macros*, SC33-6532

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
(+45) 48 14 2207 (long distance charge)	Outside North America

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Glossary

APPLYLOG. A Data Restore feature function that applies the DB2 statements, which had previously been extracted from the log (and log archives) during the RELOAD operation with RECOVERY=YES.

archive. Generic term for a backup of a complete database. Determine from context whether a Data Restore archive (created by the Data Restore feature command BACKUP) or an DB2 archive is meant. See also “online” and “offline.”

ARCHIVE. The DB2 command ARCHIVE.

backup. Generic term for backing up data. This can be either a user archive, or any of the DB2 commands DATAUNLOAD, UNLOAD, ARCHIVE or of the Data Restore feature commands UNLOAD, SELECT, BACKUP.

BACKUP. A Data Restore feature function producing a Data Restore archive.

buffer. A portion of storage used to hold input or output data temporarily.

checkpoint. A set of cleanup and recovery actions taken periodically by the database manager. Actions include writing a copy of the database to disk, recording information in the DB2 log, and freeing storage pool space.

cluster. A VSAM file in VSE. In VSE, a dbextent is implemented by a VSAM cluster.

clustered index. An index whose sequence of key values closely corresponds to the sequence of rows stored in the table to which the index belongs.

clustering index. The first index created for a table. The DB2 database manager uses it to determine the placement of subsequent rows.

commit. (1) The operation that terminates a unit of work by releasing locks so that the database changes made by that unit of work can be perceived by other processes. (2) The process that allows data changes to be made permanent. When a commit occurs, other applications can reference the just-committed data.

concurrency. The shared use of resources by multiple interactive users or application processes at the same time.

COND. CONDITION for the UNLOAD command. Identify whether the list of dbspaces is to be included or excluded for unloading.

CUU. Channel and Unit address (virtual device address).

system (DBMS)

database management. A software system that controls the logical and physical resources and facilities of a database.

database manager. A program that processes SQL statements.

Data Restore archive. Archive, taken with the Data Restore feature command BACKUP. Nearly identical is a translated DB2 archive, see “translated archive.”

Data Restore feature. A separately priced feature of DB2.

dbextent. The physical medium where database data is stored. In VM, an extent is implemented by a minidisk, in VSE by a VSAM cluster. Storage pools are composed of one or more dbextents.

dbspace. A logical allocation of space in a storage pool contained in a database. Contains one or more tables and their associated indexes.

DBSS. Database Storage Subsystem, a part of DB2.

DBSU. Database Services Utility, a part of DB2.

DDL. Data definition language.

DDR. DASD Dump Restore, a VM tool for disk backup.

deadlock. An impasse that occurs when a process is waiting for a resource that is being held by another process that is waiting for a resource currently being held by the first process.

DESCRIBE. A Data Restore feature function that produces a report about a Data Restore archive, a Data Restore translated DB2 archive or Data Restore unload file.

DML. Data manipulation language.

exec. Execute file in VM to process a sequence of commands.

EXEC. Filetype of an exec in VM.

extent. See dbextent.

FORMAT. Either a VM command or, for VSE, a Data Restore feature function to format an extent. Determine from context which one is meant.

forward log recovery. see forward recovery

forward recovery. After a backup had been taken, e.g. from a table (or a database), changes can have been made. DB2 writes these changes into the log and possibly into log archive(s). When a failure occurs and the table (or the database) has been restored, those updates must be applied to get the identical status of the table (or database) at the moment of the failure.

guest sharing. An DB2 facility that enables a user or application in a VSE, which runs as a guest operating system under VM, to access a VM DB2 database.

IBM. International Business Machines Corporation.

IDCAMS. A utility program, which is a part of VSAM in VSE.

ITSO. International Technical Support Organization.

JCL. Job Control Language, for batch processing in VSE.

KB. Kilobyte.

LISTLOG. A Data Restore feature function listing those DB2 statements that have been extracted from the log and log archive and written into work files during a previous Data Restore RELOAD operation.

lock. Mechanism used by the database manager to ensure the integrity of data. Locking prevents concurrent users from accessing inconsistent data.

log. A collection of records maintained by the DB2 database manager to describe events that occurred during the operation of the database. This information is used for recovery if a failure occurs while the database manager is executing.

log archive. A log archive is a tape or disk copy of the log, recording just the changes applied to the data since the last archive or log archive.

Logical Unit of Work (LUW). A recoverable sequence of operations within an application process. At any time, an application process is a single unit of work, but the life of an application process can involve many units of work as a result of “commit” or “rollback” operations.

MUM. A mode of operating the DB2 database manager, in which one or more users or application programs can access the database at the same time. Contrast with single user mode (SUM).

offline. While the database manager is not active. Archives that can be taken offline are DB2 archives, Data Restore feature BACKUP and user archives.

online. While the database manager is active. Archives that can be taken online are only DB2 archives.

pool. See storage pool.

QMF. Query Management Facility, an IBM program to ease SQL queries.

recover. Either a **generic term** for repairing a system, a database or data from a failure.
Or a **specific term:** to apply to a restored table those changes that had occurred after the backup copy had been taken. See forward recovery.

reload. Generic term for loading data or loading a part of a database, like a table. This can be performed with either of the DB2 commands DATALOAD, RELOAD or the Data Restore feature command RELOAD.

RELOAD. A Data Restore feature function to load one single table from either of:

- an DB2 archive
- a translated DB2 archive
- a Data Restore archive
- a Data Restore unload file

and optionally extract the applicable DB2 statements from the log (and log archives) to prepare for a later forward log recovery through Data Restore APPLYLOG.

restore. Generic term for restoring data. This can be performed either through starting the database with the parameter STARTUP=R or F, or with a user restore (like DDR) and then starting the database with the parameter STARTUP=U, or with either of the DB2 commands DATALOAD, RELOAD or the Data Restore feature commands RELOAD, RESTORE.

RESTORE. A Data Restore feature function to restore a storage pool or a complete database.

retention period. Specifies in VSE, how long a file is to be kept. 0 days means, it is not kept.

rollback. The process of restoring data changed by SQL statements to the state at its last commit point. All locks are freed. Contrast with commit.

SELECT. A Data Restore feature function to select data from DB2 tables directly out of the dbextents, bypassing the database manager.

SHOWDBS. A Data Restore feature function to execute a SHOW DBSPACE, bypassing the database manager.

DB2 archive. Archive, taken online with the DB2 command ARCHIVE or offline with the DB2 command SQLEND ARCHIVE.

SQLDBSU. SQL/DS Database Services Utility, a part of DB2.

SQL Master. An IBM program that provides a set of database administrator tools for DB2 databases within a VM or VSE environment.

storage pool. A storage pool is composed of one or more dbextents, and defines the physical space for one or more dbspaces.

SUM. A mode of operation, in which the DB2 database manager and one application run in the same virtual machine. No other application programs or users can access the database at the same time. Contrast with multiple user mode (MUM).

SYSIN. An input file for an exec in VM, containing definitions.

SYSPRINT. An output file from a VM process.

TRANSLATE. A Data Restore feature function that translates an DB2 archive into a Data Restore archive. For the difference between the results of Data Restore BACKUP and TRANSLATE see "translated archive."

translated archive. The output from a Data Restore feature TRANSLATE command applied on an DB2 archive. Nearly identical is a Data Restore archive taken with the Data Restore feature command BACKUP. The only difference is that 3 work files must be kept separately after a translation for a later Data Restore RELOAD, whereas during Data Restore BACKUP this information is written in front of the Data Restore archive file.

unit of work. See "logical unit of work (LUW)."

unload. Generic term to extract data from a database. This can be either of the DB2 commands DATAUNLOAD, UNLOAD, Data Restore feature commands SELECT, UNLOAD or Control Center commands SQLTABLE, SQLREORG.

UNLOAD. Either the DB2 DBSU command UNLOAD, or the Data Restore feature command UNLOAD. See the context to find which is meant, either in the heading, or in the words "DB2 UNLOAD" versus "Data Restore UNLOAD." DBSU UNLOAD copies data from a dbspace or table. Data Restore UNLOAD copies data from one or more dbspaces.

user archive. An archive that is taken using non-DB2 facilities.

VM. Any version and release of VM/ESA, supported by DB2 Server for VSE & VM V5R1.

VM/ESA. Virtual Machine/Enterprise Systems Architecture, a System/390 operating system.

VMDSS. VM Data Space Support, a separately priced feature of DB2.

VMPRF. VM Performance Reporting Facility, an IBM program.

VSAM. Virtual Storage Access Method, a function in VSE.

VSE. Any version and release of VSE/ESA, supported by DB2 Server for VSE & VM V5R1.

VSE/ESA. Virtual Storage Extended/Enterprise Systems Architecture, a System/390 operating system.

List of Abbreviations

CMS	Conversational Monitor System
CP	Control Program
CPU	Central Processing Unit
DASD	Direct Access Storage Device
DB2	DATABASE 2
DBSS	Data Base Storage System
DBSU	Data Base Services Utility
DDR	DASD Dump Restore
DFSMS	Data Facility Storage Management Subsystem
DRF	Data Reduction Facility
IBM	International Business Machines
IDCAMS	VSAM Access Method Services
ISQL	Interactive Structured Query Language
ITSO	International Technical Support Organization
IUCV	Inter-User Communication Vehicle
JCL	Job Control Language
LUW	Logical Units of Work
SCP	System Control Program
SQL/DS	Structured Query Language/Data System
VM	Virtual Machine
VM/ESA	Virtual Machine/Enterprise Systems Architecture
VMDSS	VM Data Spaces Support
VSE	Virtual Storage Extended
VSE/ESA	Virtual Storage Extended/Enterprise Systems Architecture
VSE/VSAM	Virtual Storage Extended/Virtual Storage Access Method

Index

E

F

G

H

I

T

U

V

W

ITSO Redbook Evaluation

DB2 Recovery on VSE and VM Using the Data Restore Feature
SG24-2039-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redeval@vnet.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

_____

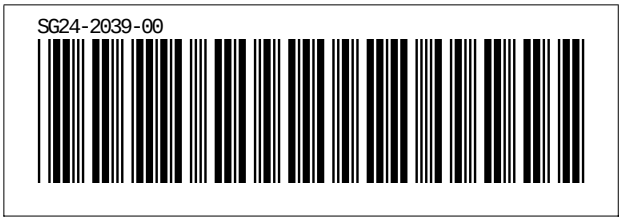
_____

_____

_____

_____

Printed in U.S.A.



Artwork Definitions

id	File	Page	References
ITLOGO	2039SU	i	i
ITLOGOS	2039SU	i	

Table Definitions

id	File	Page	References
OVER3	2039CH2	23	23
RESTAB3	2039CH2	23	23
BREAK4	2039CH2	24	24
OVERVI4	2039CH2	24	24
RESTAB4	2039CH2	24	24
BREAK2	2039CH2	24	24, 24, 24, 24
OVERVI	2039CH2	24	24
RESTAB2	2039CH2	24	24
OVER2A	2039CH3	46	47
MAN	2039CH3	46	47
BREAK1A	2039CH3	48	48, 48, 48, 48, 48, 48, 48
OVERVW	2039CH3	48	48
RESTA1A	2039CH3	48	48
R1	REDB$EVA	129	129, 129
R2	REDB$EVA	129	129

Figures

id	File	Page	References
DRF12	2039CH1	2	1
DRF13	2039CH1	4	2 4
TRANS	2039CH2	15	3 15
DESCR	2039CH2	17	4 16
REST	2039CH2	17	5 16
VSEFRM	2039CH2	18	6 18
DRFSH1C	2039CH2	19	7
SHPOOL	2039CH2	20	8
VSDREST	2039CH2	21	9 21
RSTVM	2039CH2	22	10 21
RSTVM1	2039CH2	22	11 22
DBSU1E	2039CH3	27	12

				26, 27
DBSU1F	2039CH3			
		27	13	
				26, 27
DBSU1AF	2039CH3			
		28	14	
				27
DBSU1AE	2039CH3			
		28	15	
				27
DBSUN3A	2039CH3			
		29	16	
				29
DBSUN3B	2039CH3			
		29	17	
				29
DBSR4A	2039CH3			
		29	18	
				29
DBSR4B	2039CH3			
		29	19	
				29
DRFSE1A	2039CH3			
		31	20	
				30, 31
DRFSE1B	2039CH3			
		31	21	
				31
DRFSE1C	2039CH3			
		31	22	
				30, 31
DRFSE2A	2039CH3			
		32	23	
				31
DRFSE1D	2039CH3			
		32	24	
				32
DRFSEL3	2039CH3			
		32	25	
				32
DRFUN1A	2039CH3			
		35	26	
				33, 35
DRFUN1B	2039CH3			
		35	27	
				34, 35
DRFUN1C	2039CH3			
		35	28	
				33, 35
DRFDE2A	2039CH3			
		36	29	
				36
DRFDE2B	2039CH3			
		36	30	
				36
DRFDES2	2039CH3			
		37	31	
				36, 37
DRFRE1A	2039CH3			
		39	32	
				38, 39
DRFRE1B	2039CH3			
		39	33	
				39
DRFRE1C	2039CH3			
		39	34	
				38, 39
MODIF	2039CH3			
		40	35	
				44, 46
RELPR	2039CH3			
		41	36	
				41
DRFRE2A	2039CH3			
		41	37	
				33, 41
DRFRE2B	2039CH3			
		42	38	
				33, 38, 41
DRFRE2C	2039CH3			
		43	39	
LSTLOG	2039CH3			
		44	40	
				44
LSTPRT	2039CH3			
		45	41	
				44, 46

LSTSTP	2039CH3	45	42	43
APPIN	2039CH3	46	43	46
DRF11	2039CH5	64	44	65
GRAPH0	2039CH5	69	45	69
GRAPH1	2039CH5	70	46	70
GRAPH2	2039CH5	72	47	70, 71, 71
DBCNFG	2039AX1	75	48	
DBCNFG2	2039AX1	77	49	
DBDLBL	2039AX1	77	50	
DB35VM	2039AX2	81	51	
DB51VM	2039AX2	83	52	
DB35VSE	2039AX2	84	53	86
DLBL35	2039AX2	85	54	86
JCLBKP	2039AX3	87	55	14
EXCBKP	2039AX3	87	56	14
EXCBKPS	2039AX3	87	57	14
PROJ1	2039AX3	88	58	40
PROJ2	2039AX3	88	59	41, 88
RELPRT	2039AX3	89	60	41, 41, 88
PROJ3	2039AX3	90	61	90
PROJ4	2039AX3	90	62	90
DESCVM	2039AX3	90	63	16
LSTVM	2039AX3	91	64	44
PROJNW	2039AX3	91	65	46
DLBL	2039AX3	92	66	7, 9, 39, 41, 44, 92, 92
RELOAD1	2039AX3	92	67	41, 92
RELOAD2	2039AX3	93	68	41, 93, 93
RELOA2A	2039AX3	94	69	33, 41, 93
RLDVM	2039AX3	94	70	40
JCLRST1	2039AX3	95	71	
JCLRST2	2039AX3	95	72	
RELPOL	2039AX3			

		96	73
EXCRST	2039AX3	97	74
EXCRST1	2039AX3	97	75
MAINMNU	2039AX3	98	76 97
DRM	2039AX3	99	77 98, 103
TM	2039AX3	100	78
TRANSET	2039AX3	101	79 101, 101
TRANSEL	2039AX3	102	80
TRNSLOG	2039AX3	103	81
RELOAD3	2039AX3	105	82 104
RELOAD6	2039AX3	107	83
RELOAD7	2039AX3	109	84

Headings

id	File	Page	References
INTRO	2039CH1	1	Chapter 1, Introduction to the Data Restore Feature
INTRO2	2039CH1	1	1.1, Positioning
INTRO5	2039CH1	2	1.2, DB2 Recovery Concepts
INTRO3	2039CH1	4	1.3, Functional Overview
INTRO4	2039CH1	6	1.4, Command Syntax
CHP3	2039CH2	9	Chapter 2, Archive and Recovery
ARCH1	2039CH2	9	2.1, Overview
ARCH4	2039CH2	10	2.2, User Archives
ARCH2	2039CH2	11	2.3, Archiving with DB2
ARCH21	2039CH2	12	2.4, DB2 Restore
CH3FREC	2039CH2	13	2.5, Log Recovery 20, 23, 41, 46, 55, 58, 63
ARCH3	2039CH2	13	2.6, Data Restore BACKUP 60
ARCH5	2039CH2	14	2.7, Data Restore TRANSLATE 13, 36
ARCH6	2039CH2	16	2.8, Data Restore DESCRIBE
ARCH42	2039CH2	17	2.9, Data Restore FORMAT
ARCH44	2039CH2	18	2.10, Data Restore SHOWDBS
ARCH45	2039CH2	19	2.11, Data Restore SHOWPOOL
ARCH43	2039CH2	20	2.12, Data Restore RESTORE
ARCH7	2039CH2	23	2.13, Summary
CHP2	2039CH3	25	Chapter 3, Data Unload and Reload
DTAUNL1	2039CH3	25	3.1, Overview
DTAUNL2	2039CH3	26	3.2, DBSU DATAUNLOAD and DATALOAD
DTAUNL3	2039CH3	28	3.3, DBSU UNLOAD and RELOAD 26
DTAUNL5	2039CH3	30	3.4, Data Restore SELECT

Index Entries

Tables

Processing Options

Runtime values:

Document fileid	SG242039 SCRIPT
Document type	USERDOC
Document style	REDBOOK
Profile	EDFPRF30
Service Level	0029
SCRIPT/VS Release	4.0.0
Date	97.03.29
Time	05:19:37
Device	3820A
Number of Passes	4
Index	YES
SYSVAR D	YES
SYSVAR G	INLINE
SYSVAR S	OFFSET
SYSVAR X	YES

Formatting values used:

Annotation	NO
Cross reference listing	YES
Cross reference head prefix only	NO
Dialog	LABEL
Duplex	YES
DVCF conditions file	(none)
DVCF value 1	(none)
DVCF value 2	(none)
DVCF value 3	(none)
DVCF value 4	(none)
DVCF value 5	(none)
DVCF value 6	(none)
DVCF value 7	(none)
DVCF value 8	(none)
DVCF value 9	(none)
Explode	NO
Figure list on new page	YES
Figure/table number separation	YES
Folio-by-chapter	NO
Head 0 body text	Part
Head 1 body text	Chapter
Head 1 appendix text	Appendix
Hyphenation	NO
Justification	NO
Language	ENGL
Layout	OFF
Leader dots	YES
Master index	(none)
Partial TOC (maximum level)	4
Partial TOC (new page after)	INLINE
Print example id's	NO
Print cross reference page numbers	YES
Process value	(none)
Punctuation move characters	.,
Read cross-reference file	(none)
Running heading/footing rule	NONE
Show index entries	NO
Table of Contents (maximum level)	3
Table list on new page	YES
Title page (draft) alignment	RIGHT
Write cross-reference file	(none)

Imbed Trace

Page 0	2039SU
Page 0	2039VARS
Page 0	REDB$BOE
Page i	REDB$ED1
Page i	2039EDNO
Page i	REDB$ED2
Page xi	2039ABST
Page xi	2039ACKS
Page xii	REDB$COM
Page xii	2039IMBD
Page xii	2039CH1
Page 7	2039CH2
Page 24	2039CH3
Page 48	2039CH4
Page 56	2039CH5
Page 72	2039AX1
Page 78	2039AX2
Page 86	2039AX3
Page 111	2039SPEC
Page 111	REDB$SPE
Page 112	2039TMKS
Page 112	2039BIBL
Page 114	REDB$ORD
Page 117	2039GLOS
Page 121	2039ABRV
Page 128	REDB$EVA