DB2 Server for VSE & VM

Data Restore Guide

Version 7 Release 3

SC09-2991-01

DB2 Server for VSE & VM

Data Restore Guide

Version 7 Release 3

SC09-2991-01

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 271.

Order publications through your IBM representative or the IBM branch office serving your locality or by calling 1-800-879-2755 in the United States or 1-800-IBM-4YOU in Canada.

Contents

About This Book

This book shows how to use the Data Restore feature of the IBM DATABASE 2 for VSE and VM (DB2) Version 7 Release 3. This book contains a description of the tasks associated with the use of Data Restore in a Virtual Machine/Enterprise System Architecture (VM/ESA) environment or a Virtual Storage Extended/Extended System Architecture (VSE/ESA) environment.

Who Should Use This Book

This book is a guide and reference for users of the Data Restore feature of DB2 Server for VSE & VM Version 7 Release 3, particularly database operators and administrators who maintain DB2 Server for VSE & VM databases.

How to Use This Book

This book describes Data Restore and shows how and when to use it.

Organization

This book is organized as follows:

- "Summary of Changes" on page xiii lists changes to the product since Version 6 Release 1.
- Chapter 1, "How to Recover from Failures" on page 3 describes some of the errors you can encounter and how to recover.
- Chapter 2, "Recovery Strategies" on page 7 gives you recommendations and scenarios to choose from to define your own archive and recovery strategy.
- Chapter 3, "Archive and Recovery" on page 23 describes the options provided to back up a complete DB2 database and restore its data.
- Chapter 4, "Installing Data Restore" on page 33 provides the necessary steps for installing Data Restore in any of the following environments: VM/ESA, VSE/ESA, or VSE Guest Sharing.
- Chapter 5, "Migration" on page 67 provides the necessary steps for migrating Data Restore in any of the following environments: VM/ESA, VSE/ESA, or VSE Guest Sharing.
- Chapter 6, "How to Apply Service" on page 69 describes the steps required to apply service to Data Restore.
- Chapter 7, "Data Unload and Reload" on page 73 describes the unload and load procedures.
- Chapter 8, "Backing Up an Entire Database" on page 91 describes how to use Data Restore to back up DB2 Server for VSE & VM databases.
- Chapter 9, "Restoring an Entire Database or an Entire Storage Pool" on page 103 describes how to use Data Restore to restore an entire DB2 Server for VSE & VM database or an entire storage pool from either a user or a DB2 Server for VSE & VM archive.
- Chapter 10, "Backing Up Parts of a Database" on page 121 describes how to use Data Restore to unload dbspaces because you want to backup only a single dbspace and not the entire database.

- Chapter 11, “Restoring Logical Elements” on page 125 describes how to use Data Restore to reload a single table or a group of tables with its full environment: indexes, referential integrity, views, grants, comments, and labels.
- Chapter 12, “Accessing Data Whether the Server is Up or Down” on page 147 describes how to select data from a table while the application server is up or down.
- Chapter 13, “Displaying the Contents of an Archive File” on page 151 describes how to use Data Restore to display the list of all tables in an archive or an unload file.
- Chapter 14, “Displaying Dbspace Information” on page 155 describes how to produce a report from the application server (which can either be online or offline) to show the number of used pages (header, data, or index) in the database.
- Chapter 15, “Displaying Pool Organization” on page 161 describes how the user can produce a report from the application server to show the dbextent and storage pool organization.
- Chapter 16, “Command Syntax” on page 167 contains all the command syntax diagrams and examples.
- Appendix A, “Messages and Codes” on page 191 provides messages and codes.
- Appendix B, “Command Examples” on page 203 provides procedures to illustrate the usage of Data Restore functions.
- Appendix C, “Problem Determination” on page 255 provides problem determination information.
- Index

Prerequisites

This book assumes the following:

- If you are installing in a VM environment, that you have a working knowledge of VM Conversational Monitor System and are acquainted with CMS commands.
- If you are installing in a VSE environment, that you have a working knowledge of VSE and are acquainted with the appropriate Job Control Language used in VSE.
- If you are installing in VSE Guest Sharing, that you have a working knowledge of VM Conversational Monitor System and VSE. It is further assumed that you are acquainted with CMS commands as well as the appropriate Job Control Language used in VSE.
- That you have a working knowledge of database manager backup and recovery concepts.

Terminology

TERM	Meaning
APPLYLOG	A Data Restore function to apply the DB2 statements that were extracted from the log during the RELOAD operation.
Archive	A Data Restore archive.
Associate Full Backup	The associate full backup for an incremental backup is the most recent full backup taken before the incremental backup. For example, if a full backup is taken on a Sunday night, and incremental backups are performed each night during the week,

the associate full backup for each of the incremental backups, is the full backup taken on the Sunday night preceding the incremental backup.

backup A backup is a user archive taken with the Data Restore Feature BACKUP command. It is a copy of the entire database. This is the default function for the BACKUP command and this type of backup cannot be used as an associated backup for Incremental BACKUP. A Data Restore function to produce an archive.

BACKUP FULL
A Data Restore function to produce an archive. The produced archive could be later used as a reference for INCREMENTAL BACKUP.

BACKUP INCREMENTAL
A Data Restore function to produce an incremental archive. This function can only be used if a BACKUP FULL function has been executed. Only modified pages are saved.

Data Restore archive
A Data Restore archive. Synonym of archive.

DB2 Server for VSE & VM database archive
An DB2 Server for VSE & VM Archive produced by the ARCHIVE operator command.

DESCRIBE A Data Restore function to read an archive or unload a file and produce a report with information about the contents of the file.

FORMAT A Data Restore function to format a dbextent (VSE only).

Forward log recovery
Changes referenced in the log files are processed again after restoring tables.

FULL ARCHIVE
A database archive file produced by the BACKUP FULL function.

FULL backup A FULL backup is a user archive taken with the Data Restore feature BACKUP FULL command. It is a copy of the entire database. Additionally, BACKUP FULL initializes the database directory so that subsequent incremental backups can be taken. A full backup must be taken before an incremental backup may be taken.

INCREMENTAL ARCHIVE
A database archive file produced by the BACKUP INCREMENTAL function.

INCREMENTAL backup
An INCREMENTAL backup is a user archive taken with the Data Restore BACKUP INCREMENTAL command. It contains a copy of the pages that have been modified since the last FULL backup was taken. A full backup must be taken before an incremental archive is allowed.

LISTLOG A Data Restore function to list the contents of LMBRLG1, LMBRLG2, and LMBRLG3.

regular backup
Same as backup.

RELOAD A Data Restore function to restore parts of a database.

RESTORE A Data Restore function to restore an entire database.

SELECT A Data Restore function to access data in an application server when the database is either up or down.

SHOWDBS A Data Restore function to execute a fast show dbspace for an entire database.

SHOWPOOL A Data Restore function to list a cross reference between dbextents and storage pools organization.

SPLR Storage Pool Level Recovery is the process of restoring one or more storage pools using the POOL parameter of the RESTORE command.

TRANSLATE A Data Restore function to increase the performance when later restoring the complete database or parts of it from a DB2 archive.

UNLOAD A Data Restore function to back up parts of a database.

Unload A Data Restore unload.

VM Refers to all supported releases and versions of VM/ESA.

VSE Refers to all supported releases and versions of VSE/ESA.

Note: When these terms are mentioned in this book, they refer to the above definitions, not to DB2 Server for VSE & VM Database Services Utility functions unless explicitly mentioned.

National Language Support

The default language for Data Restore is American English.

To set a different national language, you can either:

- Specify it in your SYSIN file with the LANG parameter on the OPTIONS statement (for more information refer to "OPTIONS and CONTROL Statements" on page 167).
- Set a default value in program LMBRPARM. Refer to Chapter 4, "Installing Data Restore" on page 33.

Note: Setting the application server National Language Support does not affect the Data Restore language.

Co-existence

Data Restore Version 3 Release 5 can be used with a DB2 Server for VSE & VM Version 7 Release 3 database, however, the incremental archive and restore functions and the pool recovery function are not available. Also, DATACAPTURE setting for a given table will not be processed.

Data Restore Version 5 Release 1 can be used with a DB2 Server for VSE & VM Version 7 Release 3 database, however, the incremental archive and restore functions are not available.

Data Restore Version 7 Release 3 can be used with an SQL/DS Version 3 Release 5 database, however, the incremental archive and restore functions and the pool recovery function is not available.

Data Restore Version 7 Release 3 can be used with a DB2 Server for VSE & VM Version 5 Release 1 database, however, the incremental archive and restore functions are not available.

Summary of Changes

This is a summary of the technical changes to the DB2 Server for VSE & VM database management system for this edition of the book. Several manuals are affected by some or all of the changes discussed here. For your convenience, the changes made in this edition are identified in the text by a vertical bar (|) in the left margin. This edition may also include minor corrections and editorial changes that are not identified.

This summary does not list incompatibilities between releases of the DB2 Server for VSE & VM product; see either the *DB2 Server for VSE & VM SQL Reference*, *DB2 Server for VM System Administration*, or the *DB2 Server for VSE System Administration* manuals for a discussion of incompatibilities.

Summary of Changes for DB2 Version 7 Release 3

Version 7 Release 3 of the DB2 Server for VSE & VM database management system is intended to run on the VM/ESA Version 2 Release 4 or later environment and on the Virtual Storage Extended/Enterprise Systems Architecture (VSE/ESA®) Version 2 Release 5 Modification 1 or later environment.

Enhancements, New Functions, and New Capabilities

The following have been added to DB2 Version 7 Release 3:

Cancel TCP/IP Agent in VM

With this new functionality, if the client is connected to a DB2 for VM application server via TCP/IP, and the client disconnects from the application server, the DB2 for VM application server will detect that the client has disconnected and will immediately release the resources that it had held.

Force Inactive Users

This function introduces a new DB2 Server for VM operator command, FORCE INACTIVE. The FORCE INACTIVE command will disconnect inactive users on DB2 Server for VM. Also, when SQLEND is issued on DB2 Server for VM, inactive users will be automatically disconnected so that they do not delay database shutdown. This will make the behavior of SQLEND consistent on DB2 Server for VM and on DB2 Server for VSE.

See the *DB2 Server for VSE & VM Operation* manual for further details.

CLI/JDBC V8 Redesign

If you would like to use DB2 Server for VSE & VM as a server with access by CLI/ODBC/JDBC/OLE clients, you need the support of schema functions on the DB2/VSE&VM database servers. These schema functions allow applications to get catalog information in a way that is database vendor independent. The schema functions return a standard result set to the user. In DB2 Universal Database V8.1, these functions are implemented with views and stored procedures if the views rely on SQL that is not supported on the VSE & VM platforms.

There is now a set of stored procedures that will generate the schema functions results sets without views. The result sets returned by these stored procedures correspond to the results obtained from other servers when SELECTing from the schema views.

The schema stored procedures created on VM and VSE servers are invoked internally by CLI/JDBC drivers.

See the *DB2 Server for VSE & VM Database Administration* manual for additional information.

Control Center for VSE Enhancements

The Control Center for VSE restriction of a hard coded user name and password for connecting to a database has been removed. When a Control Center for VSE transaction is initiated in CICS, Control Center will request the user ID and the password from the terminal user. A parameter card will be used for batch programs to input the user ID and password.

Control Center for VM Enhancements

Control Center enhancements include support for new operator commands, enhancements to existing operator functions, and enhancements to current initialization parameters.

Reliability, Availability, and Serviceability Improvements

Alternate Logging

A new inactive log can now take over when the active log reaches the ARCHPCT value and alternate logging is enabled, instead of immediately forcing a log archive to be taken. A log archive can then be taken at a later time, as determined by the operator. If an attempt to switch to the inactive log occurs and that inactive log has not already been archived, the operator is forced to take an archive of the inactive and active logs.

For more information, see the following DB2 Server for VSE & VM documentation:

- *DB2 Server for VSE System Administration*
- *DB2 Server for VM System Administration*
- *DB2 Server for VSE & VM Diagnosis Guide and Reference*
- *DB2 Server for VSE & VM Operation*
- *DB2 Server for VSE Program Directory*
- *DB2 Server for VM Program Directory*

Data Restore Changes

Data Restore now allows users to use the RELOAD function with RECOVERY=YES, even if an event such as a COLDLOG has broken the logical continuity of the log files. In this case, the log can now be applied until this event is reached. Alternate logging is fully supported with Data Restore for VSE & VM V7.3.

Release Empty Pages

You can now release empty pages in any dbspace on VM without having to issue DROP DBSPACE, by using a new single user mode utility called SQLRELEP. A new single user mode startup option, STARTUP=P, is introduced to enable you to release empty pages on VSE. For more information, see the *DB2 Server for VSE & VM Database Administration* manual.

More Information on TCP/IP Messages

In the event of a TCP/IP server-side error, new messages will now be displayed with more information pertaining to the TCP/IP error. This information will include the input parameters of the service causing the error.

Migration Considerations

Migration is supported from SQL/DS® Version 3 and DB2 Server for VSE & VM Versions 5 and 6. Migration from SQL/DS Version 2 Release 2 or earlier releases is not supported. Refer to the *DB2 Server for VM System Administration* manual or the *DB2 Server for VSE System Administration* manual for migration considerations.

Part 1. Getting Started

This part gives a brief overview of the Data Restore feature (also known as Data Restore) and explains how to install and use it.

All enterprises want to reduce the time that the database is unavailable for productive business use and many are striving to reach an operational goal of 24 hours a day, 7 days a week. These goals can be achieved by reducing the maintenance window, maximizing the amount of available data, and hastening the restoration of the database because of unpredictable circumstances. Together, DB2 Server for VSE & VM and its Data Restore feature provide customers with increased flexibility in their archive and restore options while minimizing the maintenance window. DB2 Server for VSE & VM now provides the tools that give you more alternatives when balancing the time and resources required to perform database archives against the risk of failure and the acceptable time for recovery.

Chapter 1. How to Recover from Failures

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

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.

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.

DASD Failure

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

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 "Recover from System Failure" on page 105 for a detailed explanation of saving the log history information.

To recover from a damaged log disk, follow the procedure as described in the *DB2 Server for VSE System Administration* and *DB2 Server for VM System Administration* manuals.

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.

Data Disk Failure

"Data Recovery" on page 104 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 (dbextent) is described in the *DB2 Server for VSE System Administration* manual. The sequence of operations to replace a data disk (database minidisk) is described in the *DB2 Server for VM System Administration* manual.

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.

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.

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.

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 the section on "Special Topics in Recovery Design" in the *DB2 Server for VSE System Administration* and *DB2 Server for VM System Administration* manuals.

Chapter 2. 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. Chapter 5, “Migration” on page 67 gives you hints on how you can realize a change in your strategy.

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 page 12), 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.

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 most 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, refer to "Log Recovery" on page 26. 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.

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.

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 “Recover from System Failure” on page 105.

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 “Recover from System Failure” on page 105.

- 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 "Data Restore BACKUP" on page 27).

  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.
- You can improve the **performance** of your BACKUP using the INCREMENTAL BACKUP This will only save modified pages from a referenced archive produced by the BACKUP FULL command.

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 "Recover from System Failure" on page 105 and page 14.

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. The incremental backup can be a very efficient way of processing a BACKUP of such a database, by only saving modified pages, that represent a very small percentage of the database. 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. The Data Restore BACKUP can be FULL (the whole database is saved) or INCREMENTAL (only modified pages from the last FULL archive are saved).

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 on page 7.

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 page 14. 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 “Directory Disk Failure” on page 4.

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, refer to “Log Recovery” on page 26.

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 “Recover from System Failure” on page 105.

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.

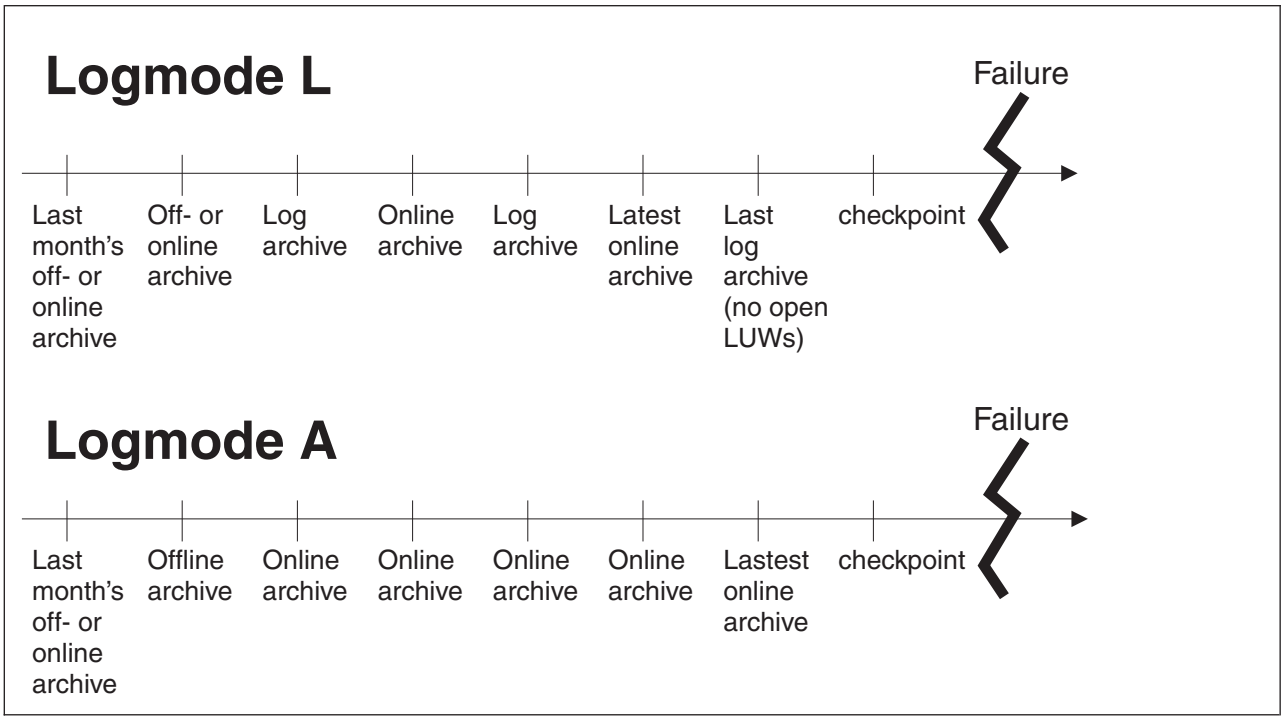


Figure 1. Sample Event Timing

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 page 15). 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 "Recover from System Failure" on page 105.

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 1 on page 14.
- 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.

Possible Strategies

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

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 change your recover strategy to one of the following.

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.

Online DB2 Archive

This strategy looks as follows:

Online DB2 ARCHIVE → *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.

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.

Data Restore BACKUP FULL/INCREMENTAL

This strategy looks as follows:

Data Restore BACKUP FULL/INCREMENTAL → *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 four 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.
- The offline archive can use the INCREMENTAL function to shorten the time necessary to produce the archive.

If Data Restore functions are processing an INCREMENTAL archive file, to reload individual tables or restore a storage pool or complete database, the associated FULL archive will be accessed while processing.

Changing Your Recovery Strategies

This section reviews some common recovery strategies and makes recommendations for new strategies to exploit new Data Restore functions.

Prior to SQL/DS Release 3.5, many users took user database archives because external tools provided better performance. To also be able to recover data at the table level, many users also performed dbspace and table unloads for critical data.

Now that the Data Restore feature is available and the database manager archive has been enhanced, there are some alternatives to be considered.

If you currently take online database archives:

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

If you used to make additional backups using 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 2.

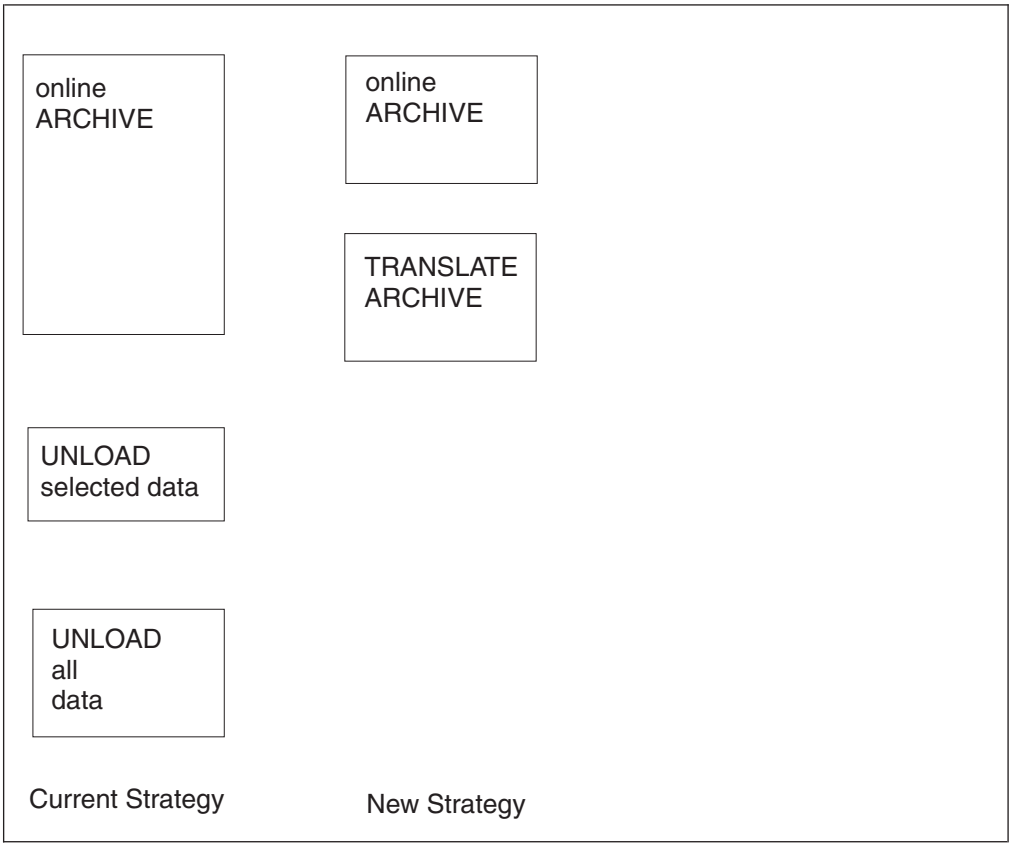


Figure 2. Migration from Online DB2 Archive

If you currently take database manager archives on shutdown:

If DB2 archive was used **offline** with SQLEND ARCHIVE, you can keep this archive option and consider the TRANSLATE, or you can recover using 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 3 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.

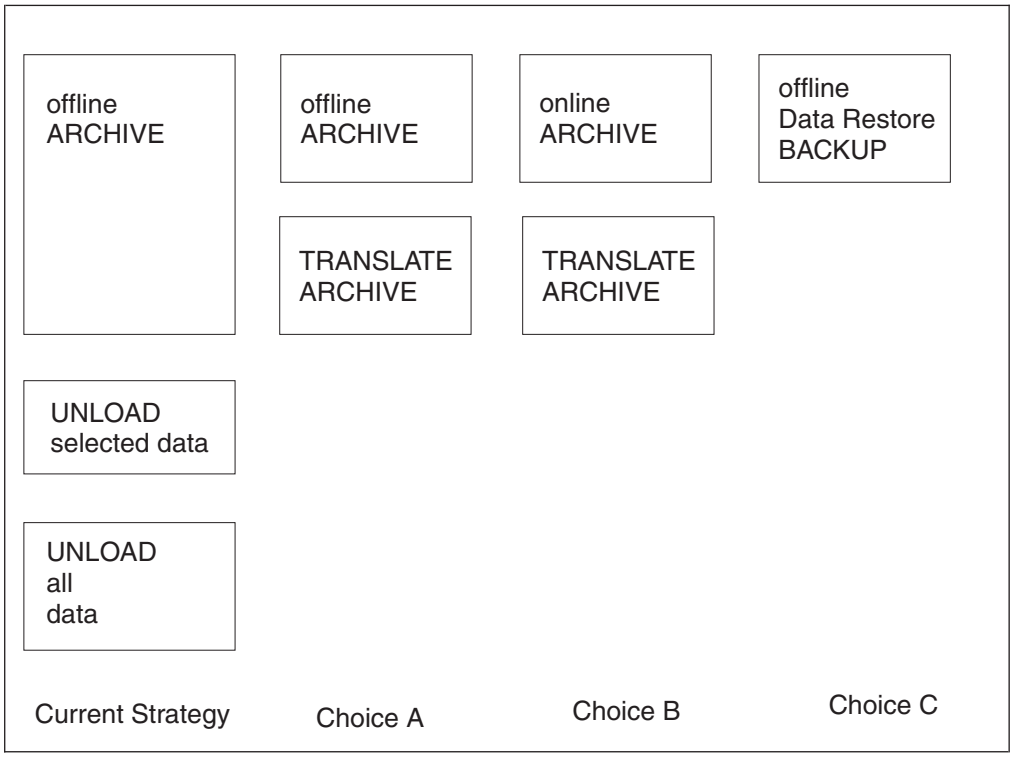


Figure 3. Migration from Offline DB2 Archive

If you currently take user database archives:

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 4 on page 22 “Current Strategy”.) 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 4 on page 22 “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, changing your recovery strategy to online DB2 ARCHIVE could be considered. The possible disadvantage is the additional translate process, which can be performed at any time. (See Figure 4 on page 22 “Choice B”.)

For recovery 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 “Recover from System Failure” on page 105, 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 14.

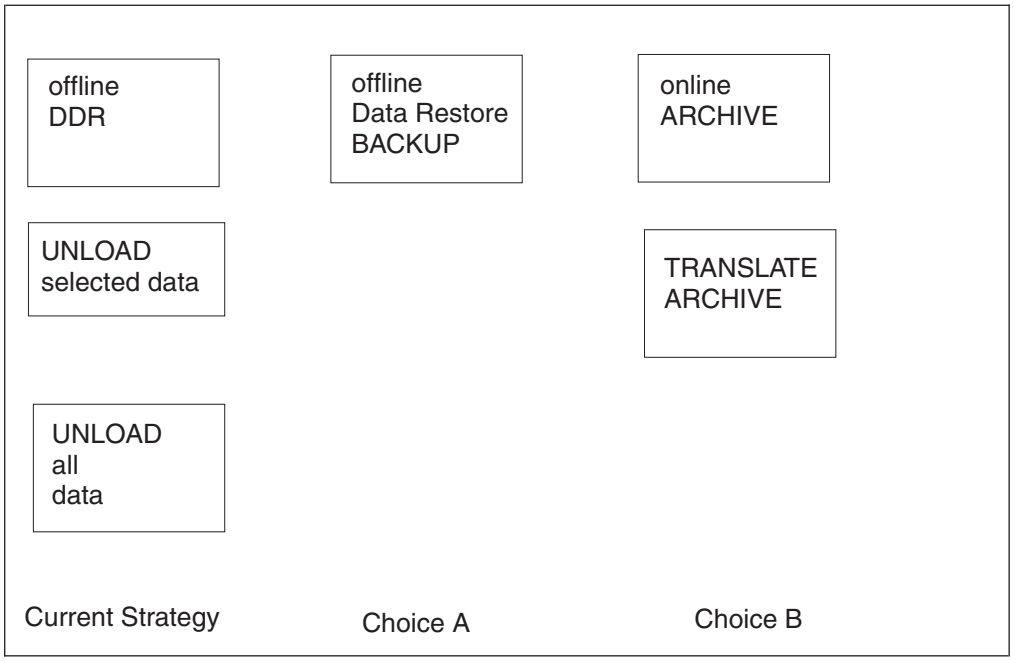


Figure 4. Migration from User Archive

Chapter 3. Archive and Recovery

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

The specifications for the work files can be found in Figure 182 on page 216.

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
- Data Restore BACKUP FULL/INCREMENTAL and RESTORE

Control Center

Control Center (formerly known as SQL Master) is a feature of DB2 Server for VSE & VM version 7. As a DB2 feature, it supports the use of Data Restore feature in a 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 also has 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, refer to "Using the TRANSLATE Function" on page 242.

For a detailed description of archive and/or backup automation refer to *DB2 for VM Control Center Operations Guide*.

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 **cannot** 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.

Archiving with DB2

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

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*.

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 Server for VSE & VM Diagnosis Guide and Reference* manual.

For a detailed description of log archives refer to the *DB2 Server for VM System Administration* and *DB2 Server for VSE System Administration* manuals.

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 the *DB2 Server for VSE System Administration* or *DB2 Server for VM System Administration* manuals.

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.

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 on page 184.

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 161 on page 203 for a sample JCL, or Figure 162 on page 203 and Figure 163 on page 203 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.

Data Restore BACKUP FULL/INCREMENTAL

Both FULL and INCREMENTAL backups are considered by the database manager as a user archive.

The BACKUP FULL archive includes the directory and dbextents, but not the log. This backup is equivalent to the one produced by the BACKUP function as explained in the previous paragraph, but can be a reference for a further INCREMENTAL backup.

The BACKUP INCREMENTAL archive includes the directory and only dbextent pages modified since the BACKUP FULL has been processed.

To process INCREMENTAL backup, a FULL BACKUP must be executed first to generate a complete copy.

The time necessary to process a BACKUP INCREMENTAL function is very low compared with any archive or user archive procedure.

While processing the INCREMENTAL BACKUP, there is no need to access any FULL archive.

In both cases, 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 on page 184.

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 161 on page 203 for a sample JCL, or Figure 162 on page 203 and Figure 163 on page 203 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.

Data Restore RESTORE

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, refer to "Log Recovery" on page 26.

The input can be any of the following:

- A Data Restore archive (the output of a Data Restore BACKUP).
- A Data Restore FULL or INCREMENTAL archive (the output of a Data Restore BACKUP FULL or BACKUP INCREMENTAL function).
- 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

  If the archive file processed has been generated by the BACKUP INCREMENTAL function, a FULL archive will be accessed to complete processing. You will be prompted to mount the right FULL archive).
- **FULLARC** - Data Restore FULL archive file necessary to complete the RESTORE, if the ARCHIV file is an INCREMENTAL BACKUP.

  Note: If the FULLARC file is used, the DEVICE2 parameter on the OPTIONS statement, should specify TAPE or DASD.

Example

Figure 5 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)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFRESTR               RESTORE DATABASE WITH DRF from tape
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
// EXEC PROC=SQLVSE02
// EXEC PROC=XTS9DLBL
// TLBL     ARCHIV
// ASSGN    SYS006,181
// MTC      REW,SYS006
/* PAUSE for monitor
// EXEC XTS91001,SIZE=AUTO
CONTROL DBNAME=SQLVSE02
RESTORE
/*
/&
* $$ EOJ
```

Figure 5. JCL File for Data Restore RESTORE

Figure 6 on page 31 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 6. EXEC File for Data Restore RESTORE from Data Restore BACKUP (RESTORE EXEC)

Figure 7 shows a sample of the EXEC to use Data Restore RESTORE.

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

Figure 7. SYSIN File for Data Restore RESTORE from Data Restore BACKUP (RESTORE SYSIN)

Summary

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**

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 “Log Recovery” on page 26.

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

Table 2. Possibilities to Archive

		Database Manager	**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

		Database Manager	**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 4. Installing Data Restore

This chapter describes the steps required to install Data Restore.

Supported Environments

- DB2 Server for VSE & VM Version 7 Release 3
- All VSE/ESA and VM/ESA releases supported by DB2 Server for VSE & VM

Most of the Data Restore functions work on physical pages. When the database manager is in a VM environment, Data Restore processes all functions in VM, even if VSE Guest Sharing is used. When the database manager is in a VSE environment, Data Restore processes all commands in VSE.

The only exception to this is the RELOAD function, which uses DB2 Server for VSE & VM functions to recreate tables and insert rows before recreating the physical database pages. If you have VSE Guest Sharing, Data Restore can process the RELOAD command in either environment.

You can execute the RELOAD process in single or multiple user mode.

Complete the Preinstallation Checklist

Before beginning, ensure that you have completed the following:

__ 1. Read Chapter 1, "How to Recover from Failures" on page 3.

__ 2. Installed DB2 Server for VSE & VM Version 7 Release 3 with at least one Version 7 Release 3 database available. This database is required to verify the installation of Data Restore.

__ 3. Decided whether you are installing in a VM, VSE, or VSE Guest Sharing environment.

__ 4. Provided enough space on DASD as shown in Figure 8 or Figure 9.

```
3375 :       16 CYLINDERS
3380 :       10 CYLINDERS
3390 :        9 CYLINDERS
9345 :       10 CYLINDERS
FBA  :    10000 BLOCKS
```

Figure 8. Minidisk Space Calculation for Data Restore VM Installation

```
3375 :        5 CYLINDERS
3380 :        4 CYLINDERS
3390 :        3 CYLINDERS
9345 :        4 CYLINDERS
FBA  :     2932 BLOCKS
```

Figure 9. Sublibrary Space Calculation for Data Restore VSE Installation

Note: If you are installing in a VSE Guest Sharing environment, obtain the total DASD requirement by adding the VM and VSE requirements.

__ 5. Checked with your IBM Support Center or used either Information/Access or IBMLink (ServiceLink) to see whether there is any additional Preventive Service Planning (PSP) information that you should be aware of.

__ 6. Ensured that the userid SQLDBA is defined in your database with DBA authority.

__ 7. Ensured that you have three public dbspaces available, having 6400, 1024, and 128 pages respectively.

__ 8. Verified that the server is running and that ISQL is available.

__ 9. Ensured that a tape drive is available to be attached to load Data Restore.

Section 1. Installing Under VM

The procedure for installing this feature in a VM environment has three steps. You must do these steps in order.

1. Define the requirements for VM resources.
2. Define the requirements for the database manager resources.
3. Run the supplied EXECs to complete the installation.

Step 1. Define the VM Resources

This step is mandatory.

1. Define a new VM userid for Data Restore and ensure that it can access all of the application servers. The minimum virtual storage size for this machine is 8 MB.
2. Allocate a minidisk of the required size (see Figure 10) to this userid. Format the minidisk (using the CMS FORMAT command) with a block size of 4 KB.

```
3375 :       16 CYLINDERS
3380 :       10 CYLINDERS
3390 :        9 CYLINDERS
9345 :       10 CYLINDERS
FBA  :    10000 BLOCKS
```

Figure 10. Minidisk Space Calculation for Data Restore VM Installation

3. If you are using an external security manager like RACF, make sure that this machine is authorized to access the minidisks of the application server (including the directory, log, and dbextent disks) in both read and write mode. (See "Security and Authorizing Access to the Server Minidisks (VM Only)" on page 65 for more information).
4. Verify that your machine can link to the application server's 195 (production) minidisk.

Step 2. Define the Database Manager Resources

This step is mandatory.

In this step, you define the database manager resources so that you can run Data Restore.

You need three public dbspaces:

- 6400-page dbspace which is defined for Data Restore's internal use
- 1024-page dbspace which is defined for Data Restore's internal use

- 128-page dbspace which is used to force a checkpoint to maintain data integrity if you use the UNLOAD function while the database is running.

  Note: This dbspace is not acquired during installation.

Using ISQL, enter the following commands:

```
(1) ---> SQLINIT DBNAME(dbname)
(2) ---> ISQL

(3) ---> CONNECT SQLDBA IDENTIFIED BY XXXXXXXX
(4) ---> ACQUIRE PUBLIC DBSPACE NAMED DATARFTR (PAGES=6400,PCTFREE=0,STORPOOL=n)
(5) ---> ACQUIRE PUBLIC DBSPACE NAMED DATARFT2 (PAGES=1024,PCTFREE=0,PCTINDEX=10,
         STORPOOL=n)
(6) ---> EXIT
```

Figure 11. Acquiring the Required Dbspaces

Note: With the STORPOOL parameter, you may select either a recoverable or a nonrecoverable storage pool. Use a nonrecoverable storage pool, if possible.

Statement 1 Initializes a connection to the database with the appropriate parameters.

Statement 2 Invokes the Interactive SQL environment.

Statement 3 Signs on to the application server with DBA authority. (Replace **XXXXXXXX** with the connect password for SQLDBA.)

Statement 4 Acquires a required public dbspace.

Statement 5 Acquires a required public dbspace.

Statement 6 Exits the ISQL environment.

You are now ready to complete the Data Restore installation.

Step 3. Complete the VM Installation Process

This process consists of four mandatory steps and two optional steps.

Step 3.1 Load Data Restore from Tape

This step is mandatory.

Log on to the Data Restore userid and execute the commands in Figure 12.

```
(1) ---> ACCESS cuu1 A
(2) ---> ATTACH cuu2 to * as 181
(3) ---> VMFPLC2 REW
(4) ---> VMFPLC2 LOAD
(5) ---> SQLINIT DBNAME(dbname)
```

Figure 12. Loading the Installation Files

Statement 1 Accesses the minidisk **cuu1** that was defined in step 1 (the minidisk must already be formatted).

Statement 2 Attaches tape drive **cuu2** as 181 and mounts the Data Restore tape on this drive.

Statement 3 Rewinds the tape on drive 181.

Statement 4 Loads the contents of the tape.

Statement 5 Runs the SQLINIT EXEC with the appropriate parameters to connect to the correct database.

Step 3.2 Create the Database Environment

This step is mandatory.

Run the following EXEC:

```
XTS9CRE
```

Figure 13. Creating the Database Environment

This EXEC creates the required database environment; for example, tables used internally by the product.

When prompted, enter the appropriate SQL CONNECT statement to connect to the database as user SQLDBA, or, if the VM userid has DBA authority, press the enter key.

Step 3.3 Load the Data Restore Packages

This step is mandatory.

Run the following EXEC:

```
    XTS9PREP
```

Figure 14. Loading the Data Restore Packages

This EXEC loads the Data Restore packages into the database.

When prompted, enter the appropriate SQL CONNECT statement to connect to the database as user SQLDBA, or, if the VM userid has DBA authority, press the enter key.

Step 3.4 Create the Data Restore Modules

This step is mandatory.

Run the following EXEC:

```
    XTS9GMOD
```

Figure 15. Creating the Data Restore Modules

This EXEC creates the Data Restore executable modules XTS91001, and XTS91002. These modules are used to invoke various functions such as BACKUP, for example.

Data Restore is now installed in one application server. You may want to take a backup of your database at this point.

Step 3.5 Install in Additional Application Servers

This step is optional.

If you want to install Data Restore in additional application servers on the same VM system, repeat the following steps:

- "Step 2. Define the Database Manager Resources" on page 34.
- "Step 3.2 Create the Database Environment" on page 36 which acquires the required dbspaces.
- "Step 3.3 Load the Data Restore Packages" on page 36 which creates the database environment.

Step 3.6 Define Default Values for Data Restore Parameters

This step is optional.

While using Data Restore, you can pass parameters in the SYSIN file. Instead of passing these parameters, you can modify and assemble the LMBRPARM program to specify default values for them.

- Edit the XTS9PARM ASSEMBLE program to customize the parameters shown in Figure 16, specifying the appropriate value in the LMBRPARM macro.

Notes:

1. Do not change the source for the LMBRPARM macro definition. Change statement 2 only.
2. To specify more than one parameter, separate each parameter with a comma. Assembler continuation rules apply.

```
(1) --->          MACRO
(1) --->          LMBRPARM  &NOTA=E,&CASE=M,&DEVICE=TAPE,                    *
(1) --->             &COMMIT=0,&MSGCLAS=1,&MSGDEV=3,                         *
(1) --->             &DBAPW=SQLDBAPW,&BASE=SQL/DS V3R5,&LANG=S01,&CONFIRM=YES
                  LCLB      &BIT1,&BIT2;
                  DC        CL11'&PWD'
         &BIT1    SETB      ('&DEVICE' EQ 'TAPE')
         &BIT2    SETB      ('&DEVICE' EQ 'DASD')
                  DC        AL1(&BIT1+&BIT2*2)
         &BIT1    SETB      ('&NOTA'(1,1) EQ 'E')
                  DC        AL1(&BIT1)
         &BIT1    SETB      ('&CASE'(1,1) EQ 'M')
         &BIT2    SETB      ('&CASE'(1,1) EQ 'U')
                  DC        AL1(&BIT1+&BIT2*2)
                  DC        AL4(&COMMIT)
                  DC        AL1(&MSGCLAS)
                  DC        AL1(&MSGDEV)
                  DC        CL8'&DBAPW'
                  DC        CL8'&BASE'
                  DC        CL4'&LANG'
                  DC        CL1'&CONFIRM'
                  MEND
         LMBRPARM CSECT
         *  in order to customize your default table
         ** modify the following statement specifying all required parameters
(2) --->          LMBRPARM  LANG=S001,BASE=dbname
                  END       LMBRPARM
```

Figure 16. XTS9PARM ASSEMBLE Program (Catalog Default Values)

Statement 1 Defines the macro.

Statement 2 Specify your default value for any parameter defined in the LMBRPARM MACRO (statement 1). The defaults entered here override the installation defaults.

Refer to "OPTIONS and CONTROL Statements" on page 167 which explains the OPTIONS parameters on the LMBRPARM macro.

- Assemble the modified program containing your default values and re-execute "Step 3.4 Create the Data Restore Modules" on page 36.

Starting Data Restore in a VM Environment

The CMS minidisk on which Data Restore is installed should be linked and accessed.

Note: Link the minidisk with an address different from the minidisk of the application server to avoid LINK errors during operation.

You can use a REXX EXEC like the one in Figure 17 to execute the Data Restore functions.

```
/**/
'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
'FILEDEF SYSPRINT DISK BACKUP SYSPRINT A'
'FILEDEF SYSIN DISK BACKUP SYSIN A'
'XTS91001'
```

Figure 17. Example EXEC to Execute Data Restore Functions

The SYSIN specifies which function to process. In Figure 17, the SYSIN file starts Data Restore to execute a BACKUP command in a VM environment.

Section 2. Installing Under VSE

The procedure for installing Data Restore under VSE has three steps. You must do these steps in order.

- Define the library to be installed.
- Restore the Data Restore distribution library.
- Run the supplied jobs to complete the installation.

Before using this chapter to install Data Restore in a VSE environment, refer to the installation manual for your VSE system for an overview of how to install optional programs.

Placement of Data Restore Distribution Libraries

When you use VSE/Interactive Interface (VSE/II), the job stream generated installs Data Restore in the VSE library PRD2.RCVvrm Although you can override this library, do not. All of the jobs described in this manual assume that the target library is PRD2.RCVvrm. If you decide to install Data Restore in a different library, you must change any LIBDEF statements that refer to PRD2.RCVvrm.

Machine-Readable Material

IBM distributes the machine-readable product on magnetic tape (reel or cartridge, non-stack). The format of the tape for VSE is:

File 1 Data Restore Copyright Records
File 2 Data Restore History File
File 3 Data Restore Product File
File 4 Tape Mark
File 5 Tape Mark

This tape is meant to be processed by the VSE maintain system history program (MSHP) or VSE/II. You need the following information to restore the Data Restore library:

- The tape label for Data Restore is DB2VSE.RCV.7.1.0
- The library for the product is PRD2
- The sublibrary for the feature is RCVvrm

Ensure that you have a distribution tape in the correct format for your VSE system. To verify this tape, scan it by running the example JCL shown in Figure 18.

Figure 19 shows a report from the scan job.

The scan report also shows the space requirements for different DASD types. You need this information if you have to define the library as described in "Step 1. Define the Library to be Installed" on page 40.

```
//JOB SCAN TAPE
* $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
* PLEASE MOUNT DB2.RCV.7.1.0 DISTRIBUTION TAPE ON A TAPE DRIVE
* AND ASSIGN SYS006 TO THAT TAPE DRIVE.
* THEN PRESS THE ENTER KEY WHEN READY ...
* $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
// PAUSE           PLEASE ASSIGN SYS006 NOW ...
// MTC REW,SYS006
// EXEC LIBR
   RESTORE * TAPE=SYS006 SCAN=YES
/*
/&
```

Figure 18. Scanning the Data Restore Feature Version 7.1.0 Distribution Library

```
RESTORE * TAPE=SYS006 SCAN=YES
L0591 BACKUP FILE ID = 'DB2VSE.RCV.7.1.0'
L123I HISTORY FILE FOUND ON TAPE
  L091I SUBLIBRARY XSOFT.RCV710 FOUND ON INPUT TAPE

   APPROXIMATE SPACE REQUIREMENT : 1093 LIBRARY BLOCKS
   3375 :       59 TRACKS  =     4 CYLINDERS 11 TRACKS
   3380 :       48 TRACKS  =     3 CYLINDERS  3 TRACKS
   3390 :       45 TRACKS  =     3 CYLINDERS  0 TRACKS
   9345 :       53 TRACKS  =     3 CYLINDERS  8 TRACKS
   FBA  :     2932 BLOCKS

L113I RETURN CODE OF RESTORE IS 0
```

Figure 19. Report from the Scan Job

IBM-Supplied Installation Aids

To help you install Data Restore on a VSE system, sample job control members are included on the Data Restore distribution tape. They are distributed as Z-type source members in the Data Restore sublibrary. Load these members as part of preparing for installation, as described in "Step 2. Restore the Data Restore Distribution Library" on page 42.

Depending on how you choose to install Data Restore, you may have to change some of the job control members before submitting them for execution. These

members and the changes needed are discussed in the following installation steps. You can punch these members out of the Data Restore sublibrary into VSE/ICCF or card decks for editing and re-submission, or, because most of them are short, you can type them in yourself.

Step 1. Define the Library to be Installed

This step is optional.

If the PRD2 library already exists and there is enough space in the library to contain the Data Restore feature, skip the rest of this step and go to "Step 2. Restore the Data Restore Distribution Library" on page 42.

If you want to install Data Restore in a library created in a VSE/VSAM space, you must:

1. Define the VSAM cluster for the VSE library using the VSAM utility IDCAMS.
2. Prepare DLBL statements for the library.
3. Define the VSE library and sublibrary using the library utility program LIBR.

If you want to install Data Restore in a library created in a non-VSE/VSAM space, you must:

1. Prepare DLBL and EXTENT statements for the library.
2. Define the VSE library and sublibrary using the library utility program LIBR.

Note: The IBM-supplied installation procedures and examples assume that Data Restore is installed in library PRD2, sublibrary RCVvrm If you install it elsewhere, make sure you change the job control statements as needed in all subsequent procedures and examples.

Step 1.1 Define a Library in VSE/VSAM Space

Refer to your VSAM manual for library cluster definitions. Figure 20 shows a job that can be used to define a VSE/VSAM library cluster.

```
         // JOB DEFINE VSE LIBRARY
         // EXEC IDCAMS,SIZE=AUTO
            DEFINE CLUSTER                                  -
(1) --->       (NAME(VSE.PRD2.LIBRARY)                      -
(2) --->       VOLUMES(volidn)                              -
               NONINDEXED                                   -
               RECORDFORMAT(NOCIFORMAT)                     -
               SHR(3)                                       -
(3) --->       CYL(primary secondary)                       -
(4) --->       DATA(NAME(VSE.PRD2.LIBRARY.DATA)))           -
(4) --->       CATALOG(VSAM.MASTER.CATALOG)
         /*
         /&
```

Figure 20. Defining a VSE/VSAM Library Cluster

Statement 1 Use the cluster name **VSE.PRD2.LIBRARY.** If you change it here, also change it as shown in Figure 21 on page 41.

Statement 2 Replace **volidn** with the appropriate volume identification information.

Statement 3 Use Figure 9 on page 33 to determine the minimum primary and secondary VSAM space allocations (add 25% free space to allow for maintenance).

Statement 4 The data name **VSE.PRD2.LIBRARY.DATA** and the catalog name **VSAM.MASTER.CATALOG** are provided as examples. Modify them to conform to your installation's standards.

Step 1.2 Prepare the Library DLBL Statements

To define the file names of the library that will contain the Data Restore feature, prepare DLBL and EXTENT statements for the library, based on the type (VSE/VSAM or non-VSE/VSAM). One of the job streams shown in Figure 21 stores the applicable statements into the system standard subarea of the label information area. You should also add the applicable statements to the existing standard label loading procedure used in the automated system initialization (ASI) process.

```
         // JOB ADDSTDL FOR VSE LIBRARY IN VSE/VSAM SPACE
         // OPTIONS STDLABEL=ADD
(1) ---> // DLBL PRD2,'VSE.PRD2.LIBRARY',,VSAM,CAT=IJSYSCT,DISP=(OLD,KEEP)
         /&

or

         // JOB ADDSTDL FOR VSE LIBRARY IN NON-VSE/VSAM SPACE
         // OPTION STDLABEL=ADD
(1) ---> // DLBL PRD2,'VSE.PRD2.LIBRARY'
(2) ---> // EXTENT volidn,1,0,xxx,yyy
         /*
         /&
```

Figure 21. Adding the Data Restore Library Information in the Standard Subarea

Statement 1 If you change **PRD2** to a different value, make the same change in all the JCL provided.

Statement 2 Provide the volume identification (**volidn**) and extent information (**xxx,yyy**) in tracks. Refer to the system control statements manual for your VSE operating system for information on how to code these values.

Step 1.3 Define the Library and Sublibrary

Use the job shown in Figure 22 to define a new VSE library and sublibrary (either BAM or VSAM). For more information about library definitions, refer to the system control statements manual for your VSE system.

```
        // JOB DEFINE LIBRARY & SUBLIBRARY
        // EXEC LIBR
           ON $RC>4 GOTO ENDJOB
(1) --->   DEFINE LIB=PRD2
           DEFINE SUBLIB=PRD2.RCVvrm  R=Y
        /. ENDJOB
        /*
        /&
```

Figure 22. Defining a New Library and Sublibrary for Data Restore

Statement 1 If the library PRD2 already exists, omit the DEFINE LIB statement.

Step 2. Restore the Data Restore Distribution Library

This step is mandatory.

You can restore the Data Restore distribution library in one of two ways:

- Use the VSE/Interactive Interface, Product Installation Dialog in VSE/ESA. If you use this method, you need to know that Data Restore is shipped on a **non-stacked V2 format** tape. For details on using this method, refer to the installation manual for your VSE system.
- Create the job control statements member shown in Figure 23 to restore the distribution library, mount the distribution tape, and run your modified job.

```
        // JOB INSvrRCV        INSTALL DB2 RECOVERY FEATURE
        // MTC REW,cuu         *-- REWIND TAPE
        // ASSGN SYS002,cuu1   *--AUXILIARY HISTORY FILE
        // ASSGN SYS005,UA     *--NO RESTORE TO DOSRES
        // ASSGN SYS006,cuu    *--DISTRIBUTION TAPE
        // OPTION CATAL
        // EXEC MSHP
(1) ---> INSTALL PRODUCT FROM TAPE ID='DB2VSE.RCV.7.1.0'- *--ACTUAL TAPE ID
(2) ---> PROD IN=PRD2.RCVvrm *--IDENTIFICATION OF LIBRARIES
        DEFINE HISTORY AUX EXTENT=xxx:yyy
        /*
        // MTC RUN,cuu
        /&
```

Figure 23. Job INSvrRCV (Using MSHP to Install Data Restore in VSE)

Before execution, modify the above statements as follows:

cuu Replace with the address of the tape drive.

cuu1 Replace with the address of the disk drive on which the auxiliary history file resides.

PRD2.RCVvrm
Replace with the library and sublibrary names you are using.

xxx Replace with the starting address of the auxiliary history file (in tracks or blocks).

yyy Replace with the size of the auxiliary history file (in tracks or blocks).

Notes:

1. You may receive messages from MSHP. These messages are normal and can be ignored. For more details about the MSHP control statements, refer to the messages and codes manual for your VSE operating system.
2. The remaining installation steps do not require the distribution tape.

Step 3. Install Data Restore

To install Data Restore, do the following mandatory steps. You must do them in order.

Note: Mandatory steps are marked with an asterisk (*). Optional steps are marked with a circle (Resource Adapter Change transaction (CIRC)).
Step 3.1 * Acquire the required dbspaces.
Step 3.2 * Create the required tables.

Step 3.3 * Reload the Data Restore packages.
Step 3.4 * Link-edit the Data Restore PRDI.
Step 3.5 * Define the VSAM files required for Data Restore.
Step 3.6 Resource Adapter Change transaction (CIRC) Define default values for Data Restore parameters.

Note: If a procedure within any of the steps does not execute successfully, correct the problem and rerun the entire step before continuing.

Technique used for installation:

All of the job control members used in the install process are in the Data Restore sublibrary cataloged as Z-type members. You can use the job shown in Figure 24 as an example of the JCL required to punch out the job control members.

```
// JOB PUNCH Z TYPE
* ************************************************
* PURPOSE : JOB STREAM TO PUNCH OUT MEMBER XXXXXXXX.Z
* ************************************************
// EXEC LIBR,PARM='MSHP'
   ACCESS SUBLIB=PRD2.RCVvrm
   PUNCH XXXXXXXX.Z
/*
/&
```

Figure 24. Example of JCL Required to Punch Out Z-TYPE Members

You may have to modify several job control statements before execution as specified in the following steps.

Step 3.1 Acquire the Required Dbspaces

This step is mandatory.

You need three public dbspaces:

- 6400-page dbspace which is defined for Data Restore's internal use
- 1024-page dbspace which is defined for Data Restore's internal use
- 128-page dbspace which is used to force a checkpoint to maintain data integrity if you use the UNLOAD function while the database is running.

  Note: This dbspace is not acquired during installation.

To acquire the dbspaces needed by Data Restore, run the job control member XTS9ACQ as shown in Figure 25 on page 44. This job should end with a DB2 Server for VSE & VM return code of 0. Ensure that the job is run successfully before continuing.

```
(1) ---> ..* $$ JOB JNM=XTS9ACQ,CLASS=0
         // JOB XTS9ACQ
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         * XTS9ACQ : Data Restore FEATURE
         *         : ACQUIRING DBSPACES
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
(2) ---> // LIBDEF *,SEARCH=PRD2.DB2vrm
(3) ---> // EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(dbname)'
(4) --->  CONNECT SQLDBA IDENTIFIED BY XXXXXXXX;
(5) --->  ACQUIRE PUBLIC DBSPACE NAMED DATARFTR(PAGES=6400,PCTFREE=0,STORPOOL=xx);
(5) --->  ACQUIRE PUBLIC DBSPACE NAMED DATARFT2(PAGES=1024,PCTFREE=0,STORPOOL=xx,
                                                PCTINDEX=10);
         /*
         /&
         ..* $$ EOJ
```

Figure 25. Job XTS9ACQ (Acquiring Dbspaces)

Statement 1 Alter the POWER statement to run the job in an appropriate class.

Statement 2 Specifies the DB2 R710 library.

Statement 3 Runs program ARIDBS (the database services utility) to acquire the required dbspaces. Replace **dbname** with the appropriate database name or remove the PARM= specification if you are accessing the default database.

Statement 4 Specify the connect password for userid SQLDBA. SQLDBA must be defined with DBA authority.

Statement 5 Acquires the required public dbspaces.

Note: With the STORPOOL parameter, you may select either a recoverable or a nonrecoverable storage pool. Define the dbspaces in a nonrecoverable storage pool, if possible.

Step 3.2 Create the Required Tables

This step is mandatory.

To create the Data Restore tables, run the job control member XTS9CRE shown in Table 4. Punch the member; if the user retypes it, errors may occur. The job should end with an SQLCODE of 0. Ensure the job is run successfully. If not, review the report created, correct the cause of the error, and rerun the job.

Table 4. Job XTS9CRE (Create Environment)

```
(1) ---> ..* $$ JOB JNM=XTS9CRE,CLASS=0
         // JOB XTS9CRE
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         * XTS9CRE : Data Restore FEATURE
         *         : CREATE DATA BASE MANAGER ENVIRONMENT
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         * XTS9CRE : Data Restore FEATURE
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
(2) ---> // LIBDEF *,SEARCH=PRD2.DB2vrm
(3) ---> // EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(dbname)'
(4) ---> CONNECT SQLDBA IDENTIFIED BY XXXXXXXX;
         CREATE TABLE IXLNGVAR (DBNO SMALLINT NOT NULL,PAGENO INTEGER NOT NULL
```

Table 4. Job XTS9CRE (Create Environment) (continued)

```
          ,RECNO INTEGER NOT NULL) IN DATARFT2;
        ...
        COMMIT WORK;
        /*
        /&
        ..* $$ EOJ
```

Statement 1 Alter the POWER statement to run the job in an appropriate class.

Statement 2 Specifies the DB2 Server for VSE & VM R710 library.

Statement 3 Runs program ARIDBS (the database services utility) to acquire the required dbspaces. Replace **dbname** with the appropriate database name or remove the PARM= specification if you are accessing the default database.

Statement 4 Specify the connect password for userid SQLDBA. SQLDBA must be defined with DBA authority.

Step 3.3 Reload the Data Restore Packages

This step is mandatory.

To reload the Data Restore packages, run the job control member XTS9PREP shown in Figure 26. You **must** punch the job to run it. It cannot be retyped. Update the database name and password before you run the job.

The job should end with an SQLCODE of 0. Ensure the job is run successfully. If not, review the report created, correct the cause of the error, and rerun the job.

Note: If the job is run more than once in a database, an error may occur on the revoke command. This error can be ignored.

```
(1) ---> ..* $$ JOB JNM=XTS9PREP,CLASS=0
         // JOB XTS9PREP
         // LIBDEF *,SEARCH=PRD2.DB2vrm
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         * XTS9PREP: Data Restore FEATURE
         *         : RELOAD PACKAGES
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
(2) ---> // EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(dbname)'
(3) ---> CONNECT SQLDBA IDENTIFIED BY XXXXXXXX;
(4) ---> RELOAD PROGRAM(SQLDBA.LMBRP029) INFILE(SYSIPT) REPLACE;
         ...
         /*
(6) ---> GRANT RUN ON LMBRP064 TO PUBLIC;
         GRANT RUN ON XTS91002 TO PUBLIC;
         SET ERRORMODE CONTINUE;
(7) ---> REVOKE CONNECT FROM DATARFTR;
         COMMIT WORK;
         /*
         /&
         ..* $$ EOJ
```

Figure 26. Job XTS9PREP (Reload Packages)

Statement 1 Alter the POWER statement to run the job in an appropriate class.

Statement 2 Runs program ARIDBS (the database services utility) to acquire the required dbspaces. Replace **dbname** with the appropriate database name or remove the PARM= specification if you are accessing the default database.

Statement 3 Specify the connect password for userid SQLDBA. SQLDBA must be defined with DBA authority.

Statement 4 Reloads the programs.

Statement 5 The package to be reloaded (only one example is provided).

Statement 6 Grants authorization to run the programs.

Statement 7 Revokes connect authority from the userid DATARFTR.

Step 3.4 Link-edit the Data Restore PRDI

This step is mandatory.

To link-edit XTSOPRDI, run the job control member XTS9PRDI shown in Figure 27. This job should end with a return code of 0.

```
(1) ---> ..* $$ JOB JNM=XTS9PRDI,CLASS=0
         // JOB XTS9PRDI
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         * XTS9PRDI: Data Restore FEATURE
         *         : LINK EDIT XTSOPRDI
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
(2) ---> // LIBDEF PHASE,CATALOG=PRD2.RCVvrm     / rcv sublibrary
(3) ---> // LIBDEF OBJ,SEARCH=PRD2.DB2vrm        / sql/ds sublibrary
         // OPTION CATAL
(4) ---> PHASE XTSOPRDI,*
(5) ---> INCLUDE ARIPRDID
         /*
(6) ---> // EXEC LNKEDT,PARM='RMODE=24'
         /*
         /&
         ..* $$ EOJ
```

Figure 27. Job XTS9PRDI (Link-edit XTSOPRDI)

Statement 1 Alter the POWER statement to run the job in an appropriate class.

Statement 2 Catalogs the phase into the Data Restore library.

Statement 3 Specifies the DB2 Server for VSE library.

Statement 4 The executable phase is XTSOPRDI.

Statement 5 Includes ARIPRDID to be link-edited.

Statement 6 Executes the Linkage Editor.

Step 3.5 Define the VSAM Files for Data Restore

This step is mandatory.

To define the required VSAM files, run the job control member XTS9DEF shown in Figure 28 on page 47.

The job ends with a return code of 8 on first execution because the DELETE CLUSTER command is processed for clusters that do not yet exist.

```
(1) ---> ..* $$ JOB JNM=XTS9DEF,CLASS=0
         // JOB XTS9DEF
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         * XTS9DEF : Data Restore FEATURE
         *         : DEFINE VSAM FILES
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
(2) ---> // DLBL IJSYSUC,'vsesp.user.catalog',,VSAM
(3) ---> // EXEC IDCAMS,SIZE=AUTO
(4) --->  DELETE DIRWORK   PURGE CLUSTER
(5) --->  DEFINE CLUSTER(NAME(DIRWORK) CYL(06)-
(6) --->          VOLUME(vvvvvv) CISZ(512) REUSE-
                       RECSZ(505 505) SHR(2) NIXD)
               DELETE LMBRWRK   PURGE CLUSTER
(7) --->  DEFINE CLUSTER(NAME(LMBRWRK) CYL(05)-
(6) --->          VOLUME(vvvvvv) CISZ(4096) REUSE-
                RECSZ(4089 4089) SHR(2) NIXD)
               DELETE LMBRLG1   PURGE CLUSTER
(8) --->  DEFINE CLUSTER(NAME(LMBRLG1) CYL(01)-
(6) --->          VOLUME(vvvvvv)  REUSE-
                        RECSZ(4096 4096) SHR(2) NIXD)
               DELETE LMBRLG2   PURGE CLUSTER
(9) --->  DEFINE CLUSTER(NAME(LMBRLG2) CYL(01)-
(6) --->          VOLUME(vvvvvv)  REUSE-
                RECSZ(4096 4096) SHR(2) NIXD)
               DELETE LMBRLG3   PURGE CLUSTER
(10) ---> DEFINE CLUSTER(NAME(LMBRLG3) CYL(01)-
(6) --->          VOLUME(vvvvvv)  REUSE-
                RECSZ(4096 32750) SHR(2) NUMBERED)

         /*
         /&
         ..* $$ EOJ
```

Figure 28. Job XTS9DEF (Define VSAM Files)

Statement 1 Alter the POWER statement to run the job in an appropriate class.

Statement 2 Specifies the VSE/VSAM user catalog.

Statement 3 Executes IDCAMS to define the required clusters.

Statement 4 Deletes the cluster before redefining it.

Statement 5 Defines the DIRWORK cluster. Modify the CYL parameter to match the size of the database directory disk (BDISK).

Statement 6 Replace **vvvvvv** with the appropriate volume identifier.

Statement 7 Defines the LMBRWRK cluster to contain all of the active pages for the biggest dbspace to be reloaded with the RECOVERY=YES parameter, or the biggest dbspace containing a LONG column.

Statement 8 Defines LMBRLG1 to contain all of the changes executed on the database for tables to be reloaded with the RECOVERY=YES parameter. The maximum of this cluster would be the size of the log disk; however, this is rarely required.

Statement 9 Defines LMBRLG2 to contain 5 bytes per rollback LUW (logical unit of work) in the log files during the RECOVERY process.

Statement 10 Defines LMBRLG3 to contain LONGVARCHAR columns for the updated rows.

Note: You can specify a secondary allocation for all of the clusters.

Step 3.6 Define Default Values for Data Restore Parameters

This step is optional.

While using Data Restore, you can pass parameters in SYSIPT by using the OPTIONS statement. Instead of passing these parameters, you can modify and assemble the LMBRPARM program to specify default values for them.

To customize your parameter defaults, punch the XTS9PARM.Z member and specify the appropriate values in the LMBRPARM macro as shown in Figure 29 on page 49.

Notes:

1. Do not change the source for the LMBRPARM macro definition. Change statement 5 only.
2. To specify more than one parameter, separate each parameter with a comma. Assembler continuation rules apply.

```
(1) ---> ..* $$ JOB JNM=XTS9PARM,CLASS=0
         // JOB XTS9PARM
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         * XTS9PARM: Data Restore FEATURE
         *         : CATALOG DEFAULT VALUES
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
(2) ---> // LIBDEF PHASE,CATALOG=PRD2.RCVvrm
(2) ---> // LIBDEF *,SEARCH=PRD2.RCVvrm
         // OPTION CATAL
(3) --->  PHASE LMBRPARM,*
         // EXEC ASSEMBLY,SIZE=512K
(4) --->           MACRO
                   LMBRPARM  &NOTA=E,&CASE=M,&DEVICE=TAPE,                    *
                      &COMMIT=0,&MSGCLAS=1,&MSGDEV=3,                         *
                      &DBAPW=SQLDBAPW,&BASE=SQL/DS V3R5,&LANG=S001,&CONFIRM=YES
                   LCLB      &BIT1,&BIT2;
                   DC        CL11'&PWD'
           &BIT1   SETB      ('&DEVICE' EQ 'TAPE')
           &BIT2   SETB      ('&DEVICE' EQ 'DASD')
                   DC        AL1(&BIT1+&BIT2*2)
           &BIT1   SETB      ('&NOTA'(1,1) EQ 'E')
                   DC        AL1(&BIT1)
           &BIT1   SETB      ('&CASE'(1,1) EQ 'M')
           &BIT2   SETB      ('&CASE'(1,1) EQ 'U')
                   DC        AL1(&BIT1+&BIT2*2)
                   DC        AL4(&COMMIT)
                   DC        AL1(&MSGCLAS)
                   DC        AL1(&MSGDEV)
                   DC        CL8'&DBAPW'
                   DC        CL8'&BASE'
                   DC        CL4'&LANG'
                   DC        CL1'&CONFIRM'
                   MEND
           LMBRPARM CSECT
           * in order to customize your default table
           * modify the following statement specifying all required parameters
(5) --->           LMBRPARM  LANG=S001
                   END       LMBRPARM
           /*
           // EXEC LNKEDT,PARM='MSHP,RMODE=24'
           /*
           /&
           ..* $$ EOJ
```

Figure 29. Job XTS9PARM (Catalog Default Values)

Statement 1 Alter the POWER statement to run the job in an appropriate class.

Statement 2 Specifies the sublibrary for Data Restore.

Statement 3 Assembles and link-edits the LMBRPARM phase.

Statement 4 Defines the macro.

Statement 5 Specify your default values for any parameter defined in the LMBRPARM macro (statement 4). The defaults entered here override the installation defaults.

Refer to “OPTIONS and CONTROL Statements” on page 167, for additional information on parameter values.

Starting Data Restore in a VSE Environment

You can execute a job by specifying the library in which Data Restore is installed, unless it is defined in standard libraries. (You can create the jobs and store them in standard libraries for later use.)

You can define a job like the one in Figure 30 to execute the Data Restore functions in a VSE environment.

```
// JOB BACKUP
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
// EXEC PROC=ARIS73DB
// TLBL ARCHIV,'ARCHIVE.DB2'
// ASSGN SYS006,180
// MTC REW,SYS006
// EXEC XTS91001,SIZE=AUTO
.. function to process
/*
```

Figure 30. Example JCL to Execute Data Restore Functions

The SYSIN file specifies the command to process in the JCL after the EXEC XTS91001 statement.

Section 3. Installing Under VSE Guest Sharing

The procedure for installing Data Restore in a VSE Guest Sharing environment has five steps:

1. Define the requirements for VM resources.
2. Define the requirements for the database manager resources.
3. Run the supplied EXECs to complete the VM installation.
4. Define the requirements for VSE resources.
5. Run the supplied jobs to complete the VSE installation.

Note: In a VSE Guest Sharing environment, only RELOAD with RECOVERY=NO can be executed from a VSE partition. No other Data Restore functions are available since these functions need direct access (LINK and ACCESS) to the database mini-disks. It is strongly recommended that Data Restore be installed as a VM product in a VSE Guest Sharing environment.

Step 1. Define the VM Resources

This step is mandatory.

1. Define a new VM userid for Data Restore and ensure that it can access all of the application servers. The minimum virtual storage size for this machine is 8 MB.
2. Allocate a minidisk of the required size (see Figure 31 on page 51) to this userid. Format the minidisk (using the CMS FORMAT command) with a block size of 4 KB.

```
3375 :       16 CYLINDERS
3380 :       10 CYLINDERS
3390 :        9 CYLINDERS
9345 :       10 CYLINDERS
FBA  :    10000 BLOCKS
```

Figure 31. Minidisk Space Calculation for Data Restore VM Installation

3. If you are using an external security manager like RACF, make sure that this machine is authorized to access the minidisks of the application server (including the directory, log, and dbextent disks) in both read and write mode. (See "Security and Authorizing Access to the Server Minidisks (VM Only)" on page 65 for more information).
4. Verify that your machine can link to the application server's 195 (production) minidisk.

Step 2. Define the Database Manager Resources

This step is mandatory.

In this step, you define the resources in the database manager so that you can run Data Restore.

You need three public dbspaces:

- 6400-page dbspace which is defined for Data Restore's internal use
- 1024-page dbspace which is defined for Data Restore's internal use
- 128-page dbspace which is used to force a checkpoint to maintain data integrity if you use the UNLOAD function while the database is running.

  Note: This dbspace is not acquired during installation.

Using ISQL, enter the following commands:

```
(1) ---> SQLINIT DBNAME(dbname)
(2) ---> ISQL

(3) ---> CONNECT SQLDBA IDENTIFIED BY XXXXXXXX
(4) ---> ACQUIRE PUBLIC DBSPACE NAMED DATARFTR (PAGES=6400,PCTFREE=0,STORPOOL=n)
(5) ---> ACQUIRE PUBLIC DBSPACE NAMED DATARFT2 (PAGES=1024,PCTFREE=0,PCTINDEX=10,
         STORPOOL=n)
(6) ---> EXIT
```

Figure 32. Acquiring the Required Dbspaces.

Note: With the STORPOOL parameter, you may select either a recoverable or a nonrecoverable storage pool. Use a nonrecoverable storage pool, if possible.

Statement 1 Initializes a connection to the database with the appropriate parameters.

Statement 2 Invokes the Interactive SQL environment.

Statement 3 Signs on to the application server with DBA authority. (Replace **XXXXXXXX** with the connect password for SQLDBA.)

Statement 4 Acquires a required public dbspace.

Statement 5 Acquires a required public dbspace.

Statement 6 Exits the ISQL environment.

You are now ready to complete the Data Restore installation on VM.

Step 3. Complete the VM Installation Process

This process consists of four mandatory steps and two optional steps.

Step 3.1 Load Data Restore from Tape

This step is mandatory.

Log on to the Data Restore userid and execute the commands in Figure 33.

```
(1) ---> ACCESS cuu1 A
(2) ---> ATTACH cuu2 to * as 181
(3) ---> VMFPLC2 REW
(4) ---> VMFPLC2 LOAD
(5) ---> SQLINIT DBNAME(dbname)
```

Figure 33. Loading the Installation Files

Statement 1 Accesses the minidisk **cuu1** that was defined in step 1 (the minidisk must already be formatted).

Statement 2 Attaches tape drive **cuu2** as 181 and mounts the Data Restore tape on this drive.

Statement 3 Rewinds the tape on drive 181.

Statement 4 Loads the contents of the tape.

Statement 5 Runs the SQLINIT EXEC with the appropriate parameters to connect to the correct database.

Step 3.2 Create the Database Environment

This step is mandatory.

Run the following EXEC:

```
XTS9CRE
```

Figure 34. Creating the Database Environment

This EXEC creates the required database environment; for example, tables used internally by the product.

When prompted, enter the appropriate SQL CONNECT statement to connect to the database as user SQLDBA, or, if the VM userid has DBA authority, press the enter key.

Step 3.3 Load the Data Restore Packages

This step is mandatory.

Run the following EXEC:

```
XTS9PREP
```

Figure 35. Loading the Data Restore Packages

This EXEC loads the Data Restore packages into the database.

When prompted, enter the appropriate SQL CONNECT statement to connect to the database as user SQLDBA, or, if the VM userid has DBA authority, press the enter key.

Step 3.4 Create the Data Restore Modules

This step is mandatory.

Run the following EXEC:

```
XTS9GMOD
```

Figure 36. Creating the Data Restore Modules

This EXEC creates the Data Restore executable modules XTS91001, and XTS91002. These modules are used to invoke various functions such as BACKUP, etc.

Data Restore is now installed in one application server. You may want to take a backup of your database at this point.

Note: Refer to "Step 3.5 Install in Additional Application Servers" on page 36 if you want to install Data Restore in additional application servers in the same VM system.

Step 3.5 Define Default Values for Data Restore Parameters

This step is optional.

While using Data Restore, you can pass parameters in the SYSIN file. Instead of passing these parameters, you can modify and assemble the LMBRPARM program to specify default values for them.

- Edit the XTS9PARM ASSEMBLE program to customize the parameters shown in Figure 37 on page 54, specifying the appropriate value in the LMBRPARM macro.

Notes:

1. Do not change the source for the LMBRPARM macro definition. Change statement 2 only.
2. To specify more than one parameter, separate each parameter with a comma. Assembler continuation rules apply.

```
(1) --->          MACRO
(1) --->          LMBRPARM  &NOTA=E,&CASE=M,&PWD=A,&DEVICE=TAPE,            *
(1) --->             &COMMIT=0,&MSGCLAS=1,&MSGDEV=3,                        *
(1) --->             &DBAPW=SQLDBAPW,&BASE=SQL/DS V3R5,&LANG=S001,&CONFIRM=YES
                  LCLB      &BIT1,&BIT2;
                  DC        CL11'&PWD'
         &BIT1    SETB      ('&DEVICE' EQ 'TAPE')
         &BIT2    SETB      ('&DEVICE' EQ 'DASD')
                  DC        AL1(&BIT1+&BIT2*2)
         &BIT1    SETB      ('&NOTA'(1,1) EQ 'E')
                  DC        AL1(&BIT1)
         &BIT1    SETB      ('&CASE'(1,1) EQ 'M')
         &BIT2    SETB      ('&CASE'(1,1) EQ 'U')
                  DC        AL1(&BIT1+&BIT2*2)
                  DC        AL4(&COMMIT)
                  DC        AL1(&MSGCLAS)
                  DC        AL1(&MSGDEV)
                  DC        CL8'&DBAPW'
                  DC        CL8'&BASE'
                  DC        CL4'&LANG'
                  DC        CL1'&CONFIRM'
                  MEND
         LMBRPARM CSECT
         * in order to customize your default table
         ** modify the following statement specifying all required parameters
(2) --->          LMBRPARM  LANG=S001
                  END       LMBRPARM
```

Figure 37. XTS9PARM ASSEMBLE Program (Catalog Default Values)

Statement 1 Defines the macro.

Statement 2 Specify your default value for any parameter defined in the LMBRPARM MACRO (statement 1). The defaults entered here override the installation defaults.

Refer to "OPTIONS and CONTROL Statements" on page 167 for more information on parameter values.

- Assemble the modified program containing your default values and re-execute "Step 3.4 Create the Data Restore Modules" on page 53.

Step 4. Define the VSE Resources

Refer to "Section 2. Installing Under VSE" on page 38 for initial considerations for installing Data Restore on a VSE system.

Step 4.1 Define the Library to be Installed

This step is optional.

If the PRD2 library already exists, and there is enough space in the library to contain the Data Restore feature, skip the rest of this step and go to "Step 4.5 Restore the Data Restore Distribution Library" on page 56.

If you want to install Data Restore in a library created in a VSE/VSAM space, you must:

1. Define the VSAM cluster for the VSE library using the VSAM utility IDCAMS.
2. Prepare DLBL statements for the library.
3. Define the VSE library and sublibrary using the library utility program LIBR.

If you want to install Data Restore in a library created in a non-VSE/VSAM space, you must:

1. Prepare DLBL and EXTENT statements for the library.
2. Define the VSE library and sublibrary using the library utility program LIBR.

Note: The IBM-supplied installation procedures and examples assume that Data Restore is installed in library PRD2, sublibrary RCVvrm. If you install it elsewhere make sure you change the job control statements as needed in all subsequent procedures and examples.

Step 4.2 Define a Library in VSE/VSAM Space

Refer to your VSAM manual for library cluster definitions. Figure 38 shows a job that can be used to define a VSE/VSAM library cluster.

```
         // JOB DEFINE VSE LIBRARY
         // EXEC IDCAMS,SIZE=AUTO
            DEFINE CLUSTER                                  -
(1) --->       (NAME(VSE.PRD2.LIBRARY)                      -
(2) --->       VOLUMES(volidn)                              -
               NONINDEXED                                   -
               RECORDFORMAT(NOCIFORMAT)                     -
               SHR(3)                                       -
(3) --->       CYL(primary secondary)                       -
(4) --->       DATA(NAME(VSE.PRD2.LIBRARY.DATA)))           -
(4) --->       CATALOG(VSAM.MASTER.CATALOG)
         /*
         /&
```

Figure 38. Defining a VSE VSAM Library Cluster

Statement 1 Use the cluster name **VSE.PRD2.LIBRARY.** If you change it here, also change it as shown in Figure 39 on page 56.

Statement 2 Replace **volidn** with the appropriate volume identification information.

Statement 3 Use Figure 9 on page 33 to determine the minimum primary and secondary VSAM space allocations (add 25% free space to allow for maintenance).

Statement 4 The data name **VSE.PRD2.LIBRARY.DATA** and the catalog name **VSAM.MASTER.CATALOG** are provided as examples. Modify them to conform to your installation's standards.

Step 4.3 Prepare the Library DLBL Statements

To define the file names of the library that will contain the Data Restore feature, prepare DLBL and EXTENT statements for the library, based on the type (VSE/VSAM or non-VSE/VSAM). One of the job streams shown in Figure 39 on page 56 stores the applicable statements into the system standard subarea of the label information area. You should also add the applicable statements to the existing standard label loading procedure used in the automated system initialization (ASI) process.

```
In the Standard Label Area

         // JOB ADDSTDL FOR VSE LIBRARY IN VSE/VSAM SPACE
         // OPTIONS STDLABEL=ADD
(1) ---> // DLBL PRD2,'VSE.PRD2.LIBRARY',,VSAM,CAT=IJSYSCT,DISP=(OLD,KEEP)
         /&

or

         // JOB ADDSTDL FOR VSE LIBRARY IN NON-VSE/VSAM SPACE
         // OPTION STDLABEL=ADD
(1) ---> // DLBL PRD2,'VSE.PRD2.LIBRARY'
(2) ---> // EXTENT volidn,1,0,xxx,yyy
         /*
         /&
```

Figure 39. Adding the Data Restore Library Information in the Standard Subarea

Statement 1 If you change **PRD2** to a different value, make the same change in all the JCL provided.

Statement 2 Provide the volume identification (**volidn**) and extent information (**xxx,yyy**) in tracks. Refer to the system control statements manual for your VSE operating system for information on how to code these values.

Step 4.4 Define the Library and Sublibrary

Use the job shown in Figure 40 to define a new VSE library and sublibrary (either BAM or VSAM). For more information about library definitions, refer to the control statements manual for your VSE system.

```
        // JOB DEFINE LIBRARY & SUBLIBRARY
        // EXEC LIBR
           ON $RC>4 GOTO ENDJOB
(1) --->   DEFINE LIB=PRD2
           DEFINE SUBLIB=PRD2.RCVvrm  R=Y
        /.ENDJOB
        /*
        /&
```

Figure 40. Defining a New Library and Sublibrary for Data Restore

Statement 1 If the library PRD2 already exists, omit the DEFINE LIB statement.

Step 4.5 Restore the Data Restore Distribution Library

This step is mandatory.

You can restore the Data Restore distribution library in one of two ways:

- Use the VSE/Interactive Interface, Product Installation Dialog in VSE/ESA. If you use this method, you need to know that Data Restore is shipped on a **non-stacked V2 format** tape. For details on using this method, refer to the installation manual for your VSE system.
- Create the job control statements member shown in Figure 41 on page 57 to restore the distribution library, mount the distribution tape, and run your modified job.

```
// JOB INSvrRCV        INSTALL Data Restore FEATURE
// MTC REW,cuu         *-- REWIND TAPE
// ASSGN SYS002,cuu1   *--AUXILIARY HISTORY FILE
// ASSGN SYS005,UA     *--NO RESTORE TO DOSRES
// ASSGN SYS006,cuu    *--DISTRIBUTION TAPE
// OPTION CATAL
// EXEC MSHP
INSTALL PRODUCT FROM TAPE ID='DB2VSE.RCV.7.1.0'- *-ACTUAL TAPE ID
PROD IN=PRD2.RCVvrm *--IDENTIFICATION OF LIBRARIES
DEFINE HISTORY AUX EXTENT=xxx:yyy
/*
// MTC RUN,cuu
/&
```

Figure 41. Job INSvrRCV (Using MSHP to Install Data Restore in VSE)

Before execution, modify the above statements as follows:

cuu Replace with the address of the tape drive.

cuu1 Replace with the address of the disk drive on which the auxiliary history file resides.

PRD2.RCVvrm
Replace with the library and sublibrary names you are using.

xxx Replace with the starting address of the auxiliary history file (in tracks or blocks).

yyy Replace with the size of the auxiliary history file (in tracks or blocks).

Notes:

1. You may receive messages from MSHP. These messages are normal and can be ignored. For more details about the MSHP control statements, refer to the messages and codes manual for your VSE operating system.
2. The remaining installation steps do not require the distribution tape.

Step 5. Complete the VSE Installation Process

This process consists of two mandatory steps and one optional step. You must do these steps in order.

- Link-edit the Data Restore PRDI.
- Define the VSAM files for Data Restore.
- Define default values for Data Restore parameters.

Step 5.1 Link-edit the Data Restore PRDI

This step is mandatory.

To link-edit XTSOPRDI, run the job control member XTS9PRDI shown in Figure 42 on page 58. This job should end with a return code of 0.

```
(1) ---> ..* $$ JOB JNM=XTS9PRDI,CLASS=0
         // JOB XTS9PRDI
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         * XTS9PRDI: Data Restore FEATURE
         *         : LINK EDIT XTSOPRDI
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
(2) ---> // LIBDEF PHASE,CATALOG=PRD2.RCVvrm       / rcv sublibrary
(3) ---> // LIBDEF OBJ,SEARCH=PRD2.DB2vrm          / DB2 Server for VSE sublibrary
         // OPTION CATAL
(4) --->  PHASE XTSOPRDI,*
(5) --->  INCLUDE ARIPRDID
         /*
(6) ---> // EXEC LNKEDT,PARM='RMODE=24'
         /*
         /&
         ..* $$ EOJ
```

Figure 42. Job XTS9PRDI (Link-edit XTSOPRDI)

Statement 1 Alter the POWER statement to run the job in an appropriate class.

Statement 2 Catalogs the phase into the Data Restore library.

Statement 3 Specifies the DB2 Server for VSE & VM library.

Statement 4 The executable phase is XTSOPRDI.

Statement 5 Includes ARIPRDID to be link-edited.

Statement 6 Executes the Linkage Editor.

Step 5.2 Define the VSAM Files for Data Restore

This step is mandatory.

To define the required VSAM files, run the job control member XTS9DEF shown in Figure 43 on page 59.

The job ends with a return code of 8 on first execution because the DELETE CLUSTER command is processed for clusters that do not yet exist.

```
(1) ---> ..* $$ JOB JNM=XTS9DEF,CLASS=0
         // JOB XTS9DEF
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         * XTS9DEF : Data Restore FEATURE
         *         : DEFINE VSAM FILES
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
(2) ---> // DLBL IJSYSUC,'vsesp.user.catalog',,VSAM
(3) ---> // EXEC IDCAMS,SIZE=AUTO
(4) --->  DELETE DIRWORK   PURGE CLUSTER
(5) --->  DEFINE CLUSTER(NAME(DIRWORK) CYL(06)-
(6) --->          VOLUME(vvvvvv) CISZ(512) REUSE-
                       RECSZ(505 505) SHR(2) NIXD)
               DELETE LMBRWRK   PURGE CLUSTER
(7) --->  DEFINE CLUSTER(NAME(LMBRWRK) CYL(05)-
(6) --->          VOLUME(vvvvvv) CISZ(4096) REUSE-
                RECSZ(4089 4089) SHR(2) NIXD)
               DELETE LMBRLG1   PURGE CLUSTER
(8) --->  DEFINE CLUSTER(NAME(LMBRLG1) CYL(01)-
(6) --->          VOLUME(vvvvvv)  REUSE-
                        RECSZ(4096 4096) SHR(2) NIXD)
               DELETE LMBRLG2   PURGE CLUSTER
(9) --->  DEFINE CLUSTER(NAME(LMBRLG2) CYL(01)-
(6) --->          VOLUME(vvvvvv)  REUSE-
                RECSZ(4096 4096) SHR(2) NIXD)
               DELETE LMBRLG3   PURGE CLUSTER
(10) ---> DEFINE CLUSTER(NAME(LMBRLG3) CYL(01)-
(6) --->          VOLUME(vvvvvv)  REUSE-
                RECSZ(4096 32750) SHR(2) NUMBERED)

         /*
         /&
         ..* $$ EOJ
```

Figure 43. Job XTS9DEF (Define VSAM Files)

Statement 1 Alter the POWER statement to run the job in an appropriate class.

Statement 2 Specifies the VSE/VSAM user catalog.

Statement 3 Executes IDCAMS to define the required clusters.

Statement 4 Deletes the cluster before redefining it.

Statement 5 Defines the DIRWORK cluster. Modify the CYL parameter to match the size of the database directory disk (BDISK).

Statement 6 Replace **vvvvvv** with the appropriate volume identifier.

Statement 7 Define the LMBRWRK cluster to contain all of the active pages for the biggest dbspace to be reloaded with the RECOVERY=YES parameter, or the biggest dbspace containing a LONG column.

Statement 8 Define LMBRLG1 to contain all of the changes executed on the database for tables to be reloaded with the RECOVERY=YES parameter. The maximum of this cluster would be the size of the log disk; however, this is rarely required.

Statement 9 Define LMBRLG2 to contain 5 bytes per rollback LUW (logical unit of work) in the log files during the RECOVERY process.

Statement 10 Define LMBRLG3 to contain LONGVARCHAR columns for the updated rows.

Note: You can specify a secondary allocation for all of the clusters.

Step 5.3 Define Default Values for Data Restore Parameters

This step is optional.

While using Data Restore, you can pass parameters in SYSIPT by using the OPTIONS statement. Instead of passing these parameters, you can modify and assemble the LMBRPARM program to specify default values for them.

To customize your parameter defaults, punch the XTS9PARM.Z member and specify the appropriate values in the LMBRPARM macro as shown in Figure 44 on page 61.

Notes:

1. Do not change the source for the LMBRPARM macro definition. Change statement 5 only.
2. To specify more than one parameter, separate each parameter with a comma. Assembler continuation rules apply.

```
(1) ---> ..* $$ JOB JNM=XTS9PARM,CLASS=0
         // JOB XTS9PARM
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         * XTS9PARM: Data Restore FEATURE
         *         : CATALOG DEFAULT VALUES
         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
(2) ---> // LIBDEF PHASE,CATALOG=PRD2.RCVvrm
(2) ---> // LIBDEF *,SEARCH=PRD2.RCVvrm
         // OPTION CATAL
(3) --->  PHASE LMBRPARM,*
         // EXEC ASSEMBLY,SIZE=512K
(4) --->           MACRO
                   LMBRPARM  &NOTA=E,&CASE=M,&DEVICE=TAPE,                  *
                      &COMMIT=0,&MSGCLAS=1,&MSGDEV=3,                       *
                      &DBAPW=SQLDBAPW,&BASE=SQL/DS V3R5,&LANG=S001,&CONFIRM=YES
                   LCLB      BIT1,&BIT2;
                   DC        CL11'&PWD'
         &BIT1     SETB      ('&DEVICE' EQ 'TAPE')
         &BIT2     SETB      ('&DEVICE' EQ 'DASD')
                   DC        AL1(&BIT1+&BIT2*2)
         &BIT1     SETB      ('&NOTA'(1,1) EQ 'E')
                   DC        AL1(&BIT1)
         &BIT1     SETB      ('&CASE'(1,1) EQ 'M')
         &BIT2     SETB      ('&CASE'(1,1) EQ 'U')
                   DC        AL1(&BIT1+&BIT2*2)
                   DC        AL4(&COMMIT)
                   DC        AL1(&MSGCLAS)
                   DC        AL1(&MSGDEV)
                   DC        CL8'&DBAPW'
                   DC        CL8'&BASE'
                   DC        CL4'&LANG'
                   DC        CL1'&CONFIRM'
                   MEND
         LMBRPARM CSECT
         * in order to customize your default table
         * modify the following statement specifying all required parameters
(5) --->           LMBRPARM  LANG=S001
                   END       LMBRPARM
         /*
         // EXEC LNKEDT,PARM='MSHP,RMODE=24'
         /*
         /&
         ..* $$ EOJ
```

Figure 44. Job XTS9PARM (Catalog Default Values)

Statement 1 Alter the POWER statement to run the job in an appropriate class.

Statement 2 Specifies the sublibrary for Data Restore.

Statement 3 Assembles and link-edits the LMBRPARM phase.

Statement 4 Defines the macro.

Statement 5 Specify your default values for any parameter defined in the LMBRPARM macro (statement 4). The defaults entered here override the installation defaults.

Refer to "OPTIONS and CONTROL Statements" on page 167 for additional information on parameter values.

Complex Environment Considerations

Installation Considerations

The previous sections contain installation instructions for simple environments; generally, with one or more databases in a single VM or VSE system, or in a single VM/VSE Guest Sharing environment.

This section will consider more complex installations, with multiple databases in multiple operating systems on multiple processors.

A complete installation of Data Restore, as described in previous sections, is needed on every independent system which has a DB2 Server for VSE & VM server installed. Some installation steps must be repeated for every additional server on the same system. If you do not do these steps for a server, Data Restore cannot be used with that server.

If you want to be able to perform Data Restore functions against more than one server on the same system at the same time, you will need to repeat some other steps. This requires that you have one set of work areas (i.e. a userid with minidisk space under VM or a set of VSAM work files under VSE) per simultaneous use of Data Restore.

Installations which use VSE Guest Sharing and also have servers running under the same VSE system must install Data Restore on both the VM and the VSE systems.

In general:

- Each SYSTEM with a DB2 Server for VSE & VM server must install the entire Data Restore feature.
- Each ADDITIONAL server must have dbspaces, tables, and packages installed.

The following table (Table 5 on page 63) summarizes which installation steps you must perform and which ones you may need to repeat, depending on your environment and operational needs. Remember that “Step 3.6 Define Default Values for Data Restore Parameters” on page 37 is optional.

Table 5. Installation Steps Summary

Step Description:	*Do Once per System?*	*Do Once per Server?*	*Repeat if Simultaneous Usage or Multiple Defaults are needed?*
VM:			
"Step 1. Define the VM Resources" on page 34	Yes		1 3
"Step 3.1 Load Data Restore from Tape" on page 35	Yes		
"Step 3.4 Create the Data Restore Modules" on page 36	Yes		
"Step 2. Define the Database Manager Resources" on page 34		Yes	
"Step 3.2 Create the Database Environment" on page 36		Yes	
"Step 3.3 Load the Data Restore Packages" on page 36		Yes	
"Step 3.6 Define Default Values for Data Restore Parameters" on page 37	Yes		3
VSE:			
"Step 1. Define the Library to be Installed" on page 40	Yes		2 3
"Step 2. Restore the Data Restore Distribution Library" on page 42	Yes		
"Step 3.4 Link-edit the Data Restore PRDI" on page 46	Yes		
"Step 3.1 Acquire the Required Dbspaces" on page 43		Yes	
"Step 3.2 Create the Required Tables" on page 44		Yes	
"Step 3.3 Reload the Data Restore Packages" on page 45		Yes	
"Step 3.5 Define the VSAM Files for Data Restore" on page 46	Yes		Yes 2
"Step 3.6 Define Default Values for Data Restore Parameters" on page 48	Yes		3

Notes on Table 5:

1. If multiple, simultaneous usage of Data Restore is required against multiple servers, multiple userids must be defined and the files and modules must be available to all of these userids. The minidisk containing the Data Restore feature can be shared R/O among these userids, but each userid requires R/W minidisk space for work files.
2. If multiple, simultaneous usage of Data Restore is required against multiple servers, you must define multiple VSAM work files. You must either use a separate VSAM user catalog for each set of identically named work files, or you must specify different work file names in the same VSAM user catalog. In either case, changes will need to be made to the JCL to point to the correct set of work files. All jobs can share the same sublibrary containing the Data Restore feature.
3. ***For VM:*** If you want different defaults to apply, you must set up a separate minidisk for each set of defaults desired. As you can override all defaults using control statements, it is normally not worthwhile to set up multiple minidisks just to have different sets of defaults. This also requires the userid running Data Restore to link and access the correct minidisk. Note that "Step 3.4 Create the Data Restore Modules" on page 36 must be repeated, as the modules must be recreated before the defaults become active.

   For VSE: If you want different defaults to apply, you must set up a separate sublibrary for each set of defaults desired. As you can override all defaults using control statements, it is normally not worthwhile to set up multiple

sublibraries just to have different sets of defaults. This also requires JCL changes to set the correct library search chain. Note that “Step 3.4 Link-edit the Data Restore PRDI” on page 46 must be repeated, as the phases must be recreated before the defaults become active.

For both: If you decide to use multiple defaults minidisks or sublibraries, you must reinstall from the distribution tape or copy from the initial minidisk/sublibrary, redefine the defaults, and then regenerate the CMS modules or re-linkedit the VSE phases. Finally, you must set up the extra work minidisk space or VSAM files.

A Complex Environment Example

Consider the following complex example, as shown in Figure 45 on page 64. In this environment, it is assumed that the VSE-Virtual system can access the VM databases via Guest Sharing.

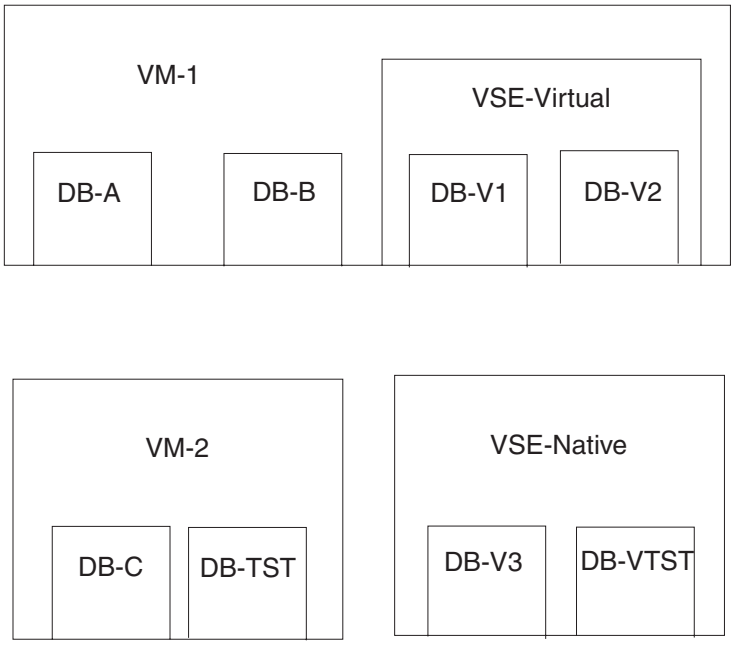


Figure 45. A Complex Environment

The recommended sequence for installing Data Restore on all servers is as follows:

1. The Data Restore feature should be installed completely into DB-TST on VM-2 by doing VM installation “Step 1. Define the VM Resources” on page 34 through “Step 3.6 Define Default Values for Data Restore Parameters” on page 37.
2. Likewise, Data Restore should be installed completely into DB-VTST on VSE-Native by doing VSE installation “Step 1. Define the Library to be Installed” on page 40 through “Step 3.6 Define Default Values for Data Restore Parameters” on page 48.
3. You can then test Data Restore under both VM and VSE to ensure it is installed and working correctly.
4. Then you would repeat VM installation steps for DB-C on VM-2:
   - “Step 2. Define the Database Manager Resources” on page 34
   - “Step 3.2 Create the Database Environment” on page 36
   - “Step 3.3 Load the Data Restore Packages” on page 36
5. Then you would repeat VSE installation steps for DB-V3 on VSE-Native:

- "Step 3.1 Acquire the Required Dbspaces" on page 43
- "Step 3.2 Create the Required Tables" on page 44
- "Step 3.3 Reload the Data Restore Packages" on page 45

6. Next, a complete install of the feature would be done for, say, DB-A on VM-1 and again for DB-V1 on VSE-Virtual.
7. Then you would repeat VM installation steps for DB-B on VM-1:
   - "Step 2. Define the Database Manager Resources" on page 34
   - "Step 3.2 Create the Database Environment" on page 36
   - "Step 3.3 Load the Data Restore Packages" on page 36
8. Then you would repeat VSE Installation steps for DB-V2 on VSE-Virtual:
   - "Step 3.1 Acquire the Required Dbspaces" on page 43
   - "Step 3.2 Create the Required Tables" on page 44
   - "Step 3.3 Reload the Data Restore Packages" on page 45

This would complete the install of Data Restore on all systems and in all servers. No extra steps are needed because of Guest Sharing access because servers also exist on the VSE-Virtual system.

If you have a Guest VSE system which accesses VM servers via VSE Guest Sharing, but does not also have DB2 Server for VSE databases, you need to define a sublibrary and install Data Restore into it, as described in "Step 4. Define the VSE Resources" on page 54 and "Step 5. Complete the VSE Installation Process" on page 57.

If simultaneous usage of Data Restore is required against multiple databases **on the same system**, you need to repeat some other steps. See the notes under Table 5 on page 63.

Security and Authorizing Access to the Server Minidisks (VM Only)

The Data Restore feature must have READ and WRITE access to the server's database disks (i.e. the directory, log, and dbextent minidisks). If you have multiple servers, this applies to these disks on **all** servers which are to be used with Data Restore.

The following possible scenarios are considered:

1. An external security manager is used

   In this case, you must authorize the Data Restore userid to have WRITE access to all the databases disks, for each database userid.
2. No external security manager, all minidisks have identical read and identical write passwords on all servers

   In this case, you must set the READPW and WRITEPW parameters to the correct passwords in the control file. These are described in "OPTIONS and CONTROL Statements" on page 167.
3. No external security manager, all minidisks do not have identical read or write passwords

   You must put LINK statements into the VM Directory entry for the Data Restore userid for all minidisks for the servers, **using the same addresses** as used by the **dbname** SQLFDEF Q file of the server. If you have multiple servers with identical addresses, this will not work, and you will need to set up multiple Data Restore userids so they can each have one or more sets of LINK commands in the VM Directory.
4. No external security manager, minidisks have no passwords

This is effectively the same case as the previous one.

Chapter 5. Migration

This chapter describes the steps required to migrate from a previous release of the Data Restore feature.

Migration under VM

Execute the following steps:

- "Step 3.1 Load Data Restore from Tape" on page 35 in VM installation
- "Step 3.3 Load the Data Restore Packages" on page 36 in VM installation
- "Step 3.4 Create the Data Restore Modules" on page 36 in VM installation
- "Step 3.5 Install in Additional Application Servers" on page 36 repeat step 3.3 only.

Migration under VSE

Execute the following steps:

- "Step 2. Restore the Data Restore Distribution Library" on page 42 in VSE installation
- "Step 3.3 Reload the Data Restore Packages" on page 45 in VSE installation
- "Step 3.4 Link-edit the Data Restore PRDI" on page 46 in VSE installation

Migration under VSE Guest Sharing

Execute the following steps:

- "Step 3.1 Load Data Restore from Tape" on page 52 in VSE Guest Sharing installation
- "Step 3.3 Load the Data Restore Packages" on page 52 in VSE Guest Sharing installation
- "Step 3.4 Create the Data Restore Modules" on page 53 in VSE Guest Sharing installation

To migrate any additional servers, execute "Step 3.3 Load the Data Restore Packages" on page 52 after executing SQLINIT to the additional server.

- "Step 4.5 Restore the Data Restore Distribution Library" on page 56 in VSE Guest Sharing installation
- "Step 5.1 Link-edit the Data Restore PRDI" on page 57 in VSE Guest Sharing installation

Chapter 6. How to Apply Service

This chapter describes the steps required to apply service to Data Restore.

Applying Service for Data Restore for VM

Follow the steps below to apply service to Data Restore VM.

1. Logon to the Data Restore userid.
2. Link and access the Data Restore code disk in Write mode.
3. Copy the files from the supplied tape to the code disk, replacing any old files.
4. Run the exec XTS9GMOD to recreate the Data Restore module.
5. Perform any additional steps that may be described in the PTF Cover Letter.

Applying Service for Data Restore for VSE

Follow the steps below to apply service to Data Restore for VSE.

1. Load the tape into the Data Restore sublibrary.
2. Edit the JCL as required, then execute.
3. Execute job XTS9PRDI to link-edit the Data Restore phase.
4. Perform any additional steps that may be described in the PTF Cover Letter.

Part 2. Using Data Restore

The part will help you use Data Restore and design your archiving strategy. The tasks in this chapter are:

- Backing up selected data in different formats to load it back into a relational database management system
- Creating your copy of your database and protecting it from system or DASD failures
- Restoring an entire database when a database is damaged
- Backing up parts of a database
- Reloading tables from a DB2 archive
- Extracting data from the database directly from the dbextents
- Displaying the contents of an archive file
- Displaying dbspace information
- Displaying storage pool organization.

Chapter 7. 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.

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.

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. 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 Server for VSE & VM 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 "DBSU UNLOAD and RELOAD" on page 75. 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.

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 *DB2 for VM Control Center Operations Guide* or *DB2 for VSE Control Center Operations Guide*.

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.

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 the *DB2 for VM Control Center Operations Guide* or *DB2 for VSE Control Center Operations Guide*.

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 "Recovery From a Logical Error" on page 125.

Files

The Data Restore SELECT 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**:

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

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.

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 54 on page 83, Figure 55 on page 83, and Figure 185 on page 217 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 46 on page 78 shows a sample.
- **SYSPRINT** file - a report that lists the SYSIN values, messages and results. Figure 48 on page 78 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 47 shows the FILEDEF definitions of the UNLOAD function.

Example

Figure 46, Figure 47, and Figure 48 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 46. 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 47. 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 VERSION 7.1.0
XTS9-309 Processing DB2 for VSE and VM version 7.1.0
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 48. SYSPRINT File of Data Restore UNLOAD (DRFUNLOA SYSPRINT)

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 produced by either BACKUP, BACKUP FULL or BACKUP INCREMENTAL function
- 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 49 on page 80 shows a sample.
- **SYSPRINT** file - Data Restore feature creates a report that lists the SYSIN values, messages and results. Figure 51 on page 81 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

    Note: If the input file is an incremental backup file, the incremental backup file will be processed first.
  - 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

If the archive file processed has been generated by the BACKUP INCREMENTAL function, a FULL archive will be accessed to complete processing. You will be prompted to mount the right FULL archive).

- **FULLARC** - Data Restore FULL archive file necessary to complete the RELOAD, if the ARCHIV file is an INCREMENTAL BACKUP.

  Note: If the FULLARC file is used, the DEVICE2 parameter on the OPTIONS statement, should specify TAPE or DASD.
- **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 55 on page 83 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 182 on page 216.

Example

Figure 49, Figure 50, and Figure 51 on page 81 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 49. 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 50. 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 VERSION 7.1.0
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 51. 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 186 on page 218 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.

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, refer to page 218.

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 174 on page 211.

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 52. 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 175 on page 212, for the SYSPRINT see Figure 176 on page 213. For the JCL used in VSE to reload a table refer to Figure 183 on page 216 and Figure 182 on page 216. 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 184 on page 217.

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 53. For the complete output of this function refer to Figure 176 on page 213. 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 “Log Recovery” on page 26.

```
XTS9-202 Processing database manager 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 53. Forward Recovery Prompt during Data Restore RELOAD

Figure 54 on page 83 and Figure 55 on page 83 show how to reload two tables with RECOVERY=YES specified in VM; for VSE, see Figure 185 on page 217.

```
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 54. 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 55. 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 56. SYSPRINT File for Data Restore RELOAD Tables and RECOVERY=YES

Data Restore RELOAD for Incremental Backup

If a RELOAD with RECOVERY=YES function requires multiple log archive tapes to be mounted, and the tapes are managed by a tape manager facility. During RELOAD processing, the LARCHIV filedef defines the tape file to be used. When Data Restore is finished reading a log archive tape and is preparing to read the next log archive tape, it may be necessary to modify the VOLSER for the LABELDEF and/or FILEDEF for the next log archive tapes. Data Restore will execute the ″XTS9X001 EXEC″ before attempting to OPEN a log archive tape file. For example, the XTS9X001 EXEC may be coded as:

```
/*                                                                  */
/* This exec is called by Data Restore during RELOAD with           */
/* RECOVERY=YES before opening each log file to allow you to        */
/* modify the LABELDEF and/or FILEDEF statement                     */
/*                                                                  */
/* Each time Data Restore needs to process a new tape, customers    */
/* who use a tape management facility, may need to modify the       */
/* the FILEDEF and/or LABELDEF statement to process several tapes   */
/* with different VOLSERs.                                          */
/*                                                                  */
/* You may need to modify this EXEC if you are using a tape         */
/* management system.                                               */
/*                                                                  */
/* If you are not using a tape management system, you may leave     */
/* this EXEC as it is.                                              */
*/
exit 0
/*                                                                  */
answer = '11'
do while answer = '11'
   say 'Please enter the VOLSER for the next tape'
   pull volid
   do until answer ='0' ! answer = '1' ! answer = '11'
      say 'The Next VOLSER of for ARILARCH is ' volid
      say ' Enter 0(Cancel), 1(Continue) or 11(Retry)'
      pull answer
   end
end
if answer = '0' then exit 16
'labeldef arilarch volid ' volid
exit rc
```

Figure 57. EXEC File for Data Restore During RELOAD with RECOVERY=YES

You can specify the new VOLSER for the next log archive file and a new LABELDEF is generated. The tape management facility will then know the correct tape to mount.

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 60 on page 86 shows a sample of how this message is presented to the user.

Figure 58 on page 86 shows the JCL that was used to perform this function in VSE. The JCL of XTS9DLBL is shown in Figure 182 on page 216. Figure 179 on page 214 shows the EXEC that was used to perform this function in VM.

The sample output can be found in Figure 59 on page 86. It lists the modifications shown in Figure 52 on page 82. For the file definitions, refer to “Data Restore RELOAD” on page 78.

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

Figure 58. 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 VERSION 7.1.0

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 59. 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 VERSION 7.1.0
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 60. Data Restore LISTLOG Termination

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 "Log Recovery" on page 26.

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, refer to "Data Restore RELOAD" on page 78. Figure 61 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 61. 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 181 on page 215 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 52 on page 82 and Figure 59 on page 86 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.

Summary

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 "Data Restore UNLOAD and RELOAD" on page 76.

Table 6 gives you an overview of the compatibilities among the methods to unload or reload data on a DB2 database.

Table 6. 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	-

Possibilities

Table 7 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 7. 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

Table 7. Possibilities Data Unload and Load (continued)

	DBSU				Data Restore					Control Center	
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 8. Backing Up an Entire Database

This chapter shows how to create a copy of your database to protect it from system or DASD failures.

Step 1 shows how to terminate your application server before executing the BACKUP function.

Step 2 shows how to process the backup of your database in both VM and VSE environments.

Step 3 shows how to restart your application server after processing the backup function.

Deciding How to Backup Your Database

You can make a copy of your database by taking a user archive using Data Restore. This copy can be on tape or DASD.

You can take the backup on either of 2 different device types (tape and tape, tape and disk, or disk and disk) at the same time, so that, for example, one copy can be taken offsite for archiving while the other copy is kept onsite for recovery purposes.

The file that is produced contains a copy of the directory and all dbextents that contain data if processing a BACKUP or BACKUP FULL function.

A new option can be used to produce a file that contains the directory and all dbextents pages modified since the last BACKUP FULL function. This function is the BACKUP INCREMENTAL and can save a lot of time to shorten the duration of application server shutdown.

Note: To ensure database integrity, you can only take a user archive when you have stopped the application server.

To execute the recovery process to apply all changes referenced in the log files, the database has to run in LOGMODE=A or LOGMODE=L. This will allow point-in-time recovery to be performed.

You should take a backup of each database regularly, so that if a system failure occurs and restoration is required, the time to restart the server is minimized.

For example, suppose you have taken a database backup every Friday evening, and a log archive on Tuesday and Thursday evenings, and on the next Friday, a media failure occurs. You would have to restore the last backup and reapply all changes processed on the database during the whole week using the log archives. If there was heavy activity that week, this process can take a long time.

If the backup is processed every day, and a media failure occurs, after restoring the database, only the changes made to the database on that day will have to be reapplied. Consequently, the server will be down for a shorter period of time.

Note: You must balance the time and resources required to perform the database archives against the risk of failure and the acceptable time for recovery.

You can take a database archive using either the DB2 Server for VSE & VM database archive or the Data Restore backup facility.

You can back up only the parts of the database which have been modified since the last full backup using the INCREMENTAL BACKUP function.

For example, support a full backup is executed on Sunday night, and an incremental backup is taken each day. The incremental backup produced on Monday will only contain pages that were modified on Monday. The incremental backup taken on Tuesday, will contain all pages that were modified on Monday and Tuesday (all pages modified since the last FULL BACKUP function). When a RELOAD is done using the incremental archive taken on Tuesday, the RELOAD process requires the Tuesday incremental archive tape and the associated full archive taken on Sunday. The incremental tape from Monday is not required.

Choosing DB2 Server for VSE & VM Database Archives

When you need to restore a table from an archive, you want the process to be as quick as possible.

With Data Restore, you can reload the table from the last archive taken and apply all the changes from the log files. This provides you with a quick way to recover a portion of the database (see Chapter 11, “Restoring Logical Elements” on page 125) without affecting all users of the rest of the database. The use of other unaffected tables can continue with very little or no disruption to users requiring access to the tables.

Data Restore can process DB2 Server for VSE & VM database archives directly as well as its own Data Restore archives. You will have better performance when Data Restore processes its own archives because DB2 Server for VSE & VM database archives must be processed twice instead of once. However, DB2 Server for VSE & VM database archives can be taken online while Data Restore user archives must be taken offline. If desired, a DB2 Server for VSE & VM archive may be converted to a Data Restore archive (refer to page “TRANSLATE” on page 184 for details).

For more information on the DB2 Server for VSE & VM archive and recovery procedures, refer to the *DB2 Server for VM System Administration* manual.

Backup Procedures

This section describes how to create an archive to protect your database against system failure using the BACKUP function.

Using Data Restore for Database Backups

Database backup files must be referenced in the DB2 Server for VSE & VM history area, so that if you use Data Restore to restore the database, all changes in successive log files are automatically reapplied when the STARTUP=U parameter is specified.

Step 1. Stop the DB2 Server for VSE & VM Server Before Backing Up Your Database

Before you start to backup your database, you must stop your application server.

- **If you are running the server in LOGMODE=A or L**

You must stop your server by issuing an SQLEND UARCHIVE operator command to have the backup referenced in the history area as shown in Figure 62. After all logical units of work are completed, the database manager indicates that you must save the directory and all dbextents. If the database is running in LOGMODE=L and the log is not empty, a log archive is taken before terminating the database.

When the server is stopped, take the user archive using Data Restore.

```
ARI0062A ENTER A DB2 FOR VSE OPERATOR COMMAND.
SQLEND UARCHIVE
ARI0028I THE DATABASE MANAGER IS TERMINATING.
ARI0065I OPERATOR COMMAND PROCESSING IS COMPLETE.
ARI0239I EXTERNAL LABELING OF THIS ARCHIVE IS
         TYPE    DATABASE ARCHIVE
         TIMESTAMP  09-09-95   14:30:15
ARI0205I YOU MUST USER-ARCHIVE THE DIRECTORY AND 8
                 DBEXTENTS(S) BUT NOT THE LOG
ARI0032I THE DATABASE MANAGER HAS TERMINATED.
```

Figure 62. Stopping the Server with SQLEND UARCHIVE

- **If you are running in LOGMODE=Y**

You can only recover from the archive. You cannot reapply the changes referenced in the log.

In this case, it is not necessary for the server to know when the database backup was done. You can use the SQLEND operator command to stop the server, as shown in Figure 63. If you want to ensure that the log history is updated, use the SQLEND UARCHIVE command.

```
ARI0062A ENTER A DB2 FOR VSE OPERATOR COMMAND.
SQLEND
ARI0028I THE DATABASE MANAGER IS TERMINATING.
ARI0065I OPERATOR COMMAND PROCESSING IS COMPLETED.
ARI0032I THE DATABASE MANAGER HAS TERMINATED.
```

Figure 63. Stopping the Server with SQLEND

Step 2. Processing a Data Restore Backup

To process a backup of a database, you must create a SYSIN file to identify the database you want to process and the name of the function to process. (Figure 64 on page 94 contains an overview of the BACKUP process.)

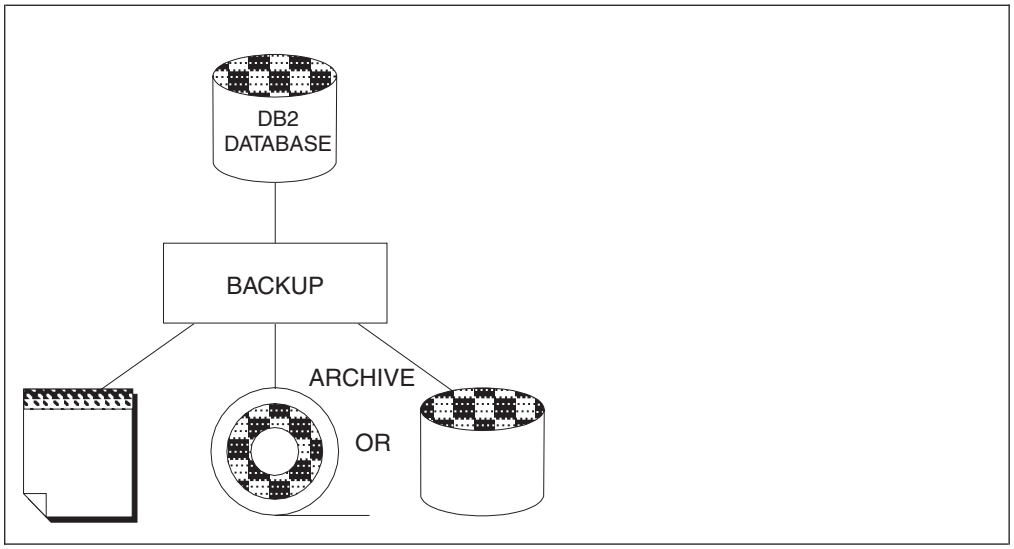


Figure 64. Overview of Backing Up a Database

In Figure 64, the DB2 Server for VSE & VM DATABASE disk represents the directory and all the defined dbextents. You can create the backup file on tape or DASD. You can also take dual backups to tape, DASD, or a combination of tape and DASD. When the backup is complete, Data Restore produces a report specifying how many blocks have been saved from the directory, how many blocks have been saved from all the saved dbextents, and a list of all the tables that can be reloaded from the file.

To back up your database, use the BACKUP command. For a incremental backup of your database, use the INCREMENTAL BACKUP command. See "BACKUP" on page 173 for more information on this command.

Step 2.1 explains how to back up your database in a VM environment. To process the BACKUP function in a VSE environment, go to "Step 2.2 Backing Up Your Database (VSE)" on page 98.

Step 2.1 Backing Up Your Database (VM)

1. Log on to the CMS machine where Data Restore is installed.
2. Verify that the SYSIN file contains a CONTROL statement with the correct DBNAME parameter and a BACKUP statement. (The DBNAME parameter must be specified to access the associated **dbname** SQLFDEF files to link the CMS minidisks to BACKUP.)

```
OPTIONS CONFIRM=NO
CONTROL DBNAME=dbname
BACKUP
```

Figure 65. Contents of SYSIN File (BACKUP SYSIN A)

3. Verify that the FILEDEFs are specified in the BACKUP EXEC and that they assign the output file to tape or DASD or both.
4. Run an EXEC to execute the XTS91001 program as shown in the example in Figure 66 on page 95. To take a dual back up, see Figure 67 on page 95.

```
        /**/
(1) ---> 'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
(2) ---> 'FILEDEF SYSPRINT DISK BACKUP SYSPRINT A'
(3) ---> 'FILEDEF SYSIN DISK BACKUP SYSIN A'
(4) ---> 'XTS91001'
```

Figure 66. Executing a BACKUP to Tape in a VM Environment.

Statement 1 Assigns the FILEDEF to a tape device. Attach the tape drive as address 181. ARCHIV is the name of the output file for the BACKUP function. The output file is variable blocked with 32760-byte blocks.

Statement 2 Creates the SYSPRINT report containing a list of all steps executed. Figure 69 on page 97 shows a function report produced by the BACKUP function in more detail.

Statement 3 Identifies the SYSIN file which contains the statements shown in Figure 65 on page 94:

You must be able able to access the **dbname** SQLFDEF files that are on the database production disk when you are executing the Data Restore functions. Make sure you link and access the minidisk where Data Restore is installed before executing the BACKUP function.

If the CONFIRM=YES parameter is specified, you must confirm that you want to continue with the backup before starting the BACKUP function.

Statement 4 Executes the BACKUP function specified in the SYSIN file.

Note: In all the other Data Restore functions, you edit the SYSIN file and modify the parameters before execution. For the BACKUP function, the procedure avoids operator intervention and does not require you to edit the file.

5. Verify the return code and list produced on SYSPRINT.

Executing a Dual Backup (VM): If you want to take a dual backup, execute an EXEC like the one shown in Figure 67 instead of the one shown in Figure 66.

```
        /**/
(1) ---> 'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
(2) ---> 'FILEDEF ARCHIV2 DISK SQLDBA BACKUP A (RECFM VB BLOCK 32760'
(3) ---> 'FILEDEF SYSPRINT DISK BACKUP SYSPRINT A'
(4) ---> 'FILEDEF SYSIN DISK BACKUP SYSIN A'
(5) ---> 'XTS91001'
```

Figure 67. Executing a Dual Backup in a VM Environment

Statement 1 Assigns the FILEDEF to a tape device. Attach the tape drive as address 181. ARCHIV is the name of the output file for the BACKUP function. The output file is variable blocked with 32760-byte blocks.

Statement 2 Assigns the FILEDEF to a disk. ARCHIV2 is the name of the output file for the BACKUP function. The output file is variable blocked with 32760-byte blocks.

Statement 3 Creates the SYSPRINT report containing a list of all processes executed. Figure 69 on page 97. shows a function report produced by the BACKUP function in more detail.

Statement 4 Identifies the SYSIN file which must contain the statements shown in Figure 68.

```
OPTIONS DEVICE2=DASD CONFIRM=NO
CONTROL DBNAME=dbname
BACKUP
```

Figure 68. Control Statement to Execute a Backup in a VM Environment

You must be able able to access the **dbname** SQLFDEF files that are on the database production disk when you are executing the Data Restore functions. Make sure you link and access the minidisk where Data Restore is installed before executing the BACKUP function.

To take a dual backup, assign a second device as DEVICE2=DASD. Figure 67 on page 95 is an example of writing one backup to tape and the other to DASD.

Note: To write dual backups to tape, assign both devices to tape. To write dual backups to DASD, assign both devices to DASD.

If the CONFIRM=YES parameter is specified, you must confirm that you want to continue with the backup before executing the backup process.

Statement 5 Executes the BACKUP function specified in the SYSIN file.

Executing a FULL or INCREMENTAL BACKUP: To process FULL backup, modify the SYSIN in Figure 66 on page 95 specifying BACKUP FULL instead of BACKUP on the last statement.

To process incremental backup, modify the SYSIN in Figure 66 on page 95 specifying BACKUP INCREMENTAL instead of BACKUP on the last statement.

```
(1)  ---> XTS9-143 OPTIONS DEVICE2=DASD CONFIRM=NO
(1)  ---> XTS9-143 CONTROL DBNAME=dbname
(1)  ---> XTS9-143 BACKUP
          XTS9-100 Data Restore feature VERSION 7.1.0
          XTS9-309 Processing DB2 Version 7 Release 3
(2)  ---> XTS9-172 DB2 was ended with LOGMODE L
(3)  ---> XTS9-141 External labeling of this archive is
          XTS9-142 Base dbname    date 17/10/95 - time 12:12:49
          XTS9-001 Processing directory
          XTS9-002           20010 Directory blocks saved
(4)  ---> XTS9-013 Table SQLDBA.COST_TABLE may be reloaded
          XTS9-013 Table SQLDBA.DEPARTMENT may be reloaded
          XTS9-013 Table SQLDBA.EMPLOYEE may be reloaded
          XTS9-013 Table SQLDBA.EMPLOYEE_ACTIVITY may be reloaded
          XTS9-013 Table SQLDBA.FOREIG may be reloaded
          XTS9-013 Table SQLDBA.INVENTORY may be reloaded
          XTS9-013 Table SQLDBA.OPERATIONS may be reloaded
          XTS9-013 Table SQLDBA.ORDERS may be reloaded
          XTS9-013 Table SQLDBA.PLAN_TABLE may be reloaded
          XTS9-013 Table SQLDBA.ROUTINE may be reloaded
          XTS9-145 Table SQLDBA.STORED QUERIES may be reloaded
          XTS9-013 Table SQLDBA.STRUCTURE_TABLE may be reloaded
          XTS9-013 Table SQLDBA.SUPPLIERS may be reloaded
          XTS9-013 Table SQLDBA.SYSLANGUAGE may be reloaded
          XTS9-013 Table SQLDBA.SYSTEXT1 may be reloaded
          XTS9-013 Table SQLDBA.SYSTEXT2 may be reloaded
          XTS9-013 Table SQLDBA.SYSUSERLIST may be reloaded
(5)  ---> XTS9-006 processing DDSK1
(5)  ---> XTS9-005              2 blocks saved
(5)  ---> XTS9-006 processing DDSK4
(5)  ---> XTS9-005              1 blocks saved
(6)  ---> XTS9-006 processing DDSK1
(7)  ---> XTS9-005           2870 blocks saved
          XTS9-006 processing DDSK2
          XTS9-005            383 blocks saved
          XTS9-006 processing DDSK3
          XTS9-005              7 blocks saved
          XTS9-006 processing DDSK4
          XTS9-005              1 blocks saved
          XTS9-006 processing DDSK5
          XTS9-005              1 blocks saved
          XTS9-006 processing DDSK6
          XTS9-005              1 blocks saved
          XTS9-006 processing DDSK7
          XTS9-005              1 blocks saved
          XTS9-006 processing DDSK8
          XTS9-005              1 blocks saved
          XTS9-007 processing successfully completed
```

Figure 69. SYSPRINT File After Executing A BACKUP

Statement 1 Displays the contents of the SYSIN file on SYSPRINT.

Statement 2 Displays the Logmode type when the server was ended.

Statement 3 Displays the external labeling of the BACKUP.

Statement 4 Displays the list of reloadable tables from the BACKUP file.

Statement 5 For some dbextents, header pages may be saved before other pages.

Statement 6 Displays each saved dbextent when processed

Statement 7 Displays the number of dbextent blocks saved is displayed.

Step 2.2 Backing Up Your Database (VSE)

1. Verify that the library where Data Restore was installed is specified on the LIBDEF statement in your backup job.
2. Verify that the SYSIN file in the JCL contains a CONTROL statement with the correct **dbname** parameter and the BACKUP statement as shown in the example in Figure 70. If you want to take a dual backup, use the example shown in Figure 71 on page 99.
3. Verify that the TLBL or DLBL statement in the JCL is correct to assign the output file to tape or DASD.
4. Run the BACKUP JCL to run the backup procedure.
5. Verify the return code and list produced on SYSLST.

```
         // JOB BACKUP
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // EXEC PROC=ARIS73DB
(3) ---> // TLBL ARCHIV,'ARCHIVE.DB2'
(4) ---> // ASSGN SYS006,180
(5) ---> // MTC REW,SYS006
(6) ---> // EXEC XTS91001,SIZE=AUTO
(7) ---> OPTIONS DEVICE=TAPE
(8) ---> CONTROL DBNAME=dbname
(9) ---> BACKUP
/*
```

Figure 70. JCL to Execute a BACKUP to Tape in a VSE Environment

Statement 1 Specifies the DB2 and Data Restore libraries.

Statement 2 Executes the procedure containing all the DB2 DLBLs.

Statement 3 Specifies the ddname of the output file. In this example ARCHIV is the *ddname* of the output file for the BACKUP function.

Statement 4 Assigns the file to a tape device. The output file is variable blocked with 32760 bytes' blocks.

Statement 5 Positions the tape at the beginning.

Statement 6 Runs the program to execute the BACKUP function.

Statement 7 Specifies the device.

Statement 8 Uses the control statement to specify the name of the database. This **dbname** is only used to name the output file.

Statement 9 Processes the BACKUP function.

The SYSLST report contains a list of all the executed processes. A function report produced by the BACKUP function is described in Figure 72 on page 100.

Executing a Dual Backup in the VSE Environment: If you want to take a dual backup, execute a BACKUP job using JCL similar to that shown in Figure 71 on page 99 instead of that shown in Figure 70.

```
          // JOB BACKUP
(1)  ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2)  ---> // EXEC PROC=ARIS73DB
(3)  ---> // TLBL ARCHIV,'ARCHIVE.DB2'
(4)  ---> // ASSGN SYS006,180
(5)  ---> // MTC REW,SYS006
(6)  ---> // DLBL ARCHIV2,'ARCHIVE.DB2',,VSAM,RECSIZE=32750,RECORDS=(xxx,yyy)
(7)  ---> // EXEC XTS91001,SIZE=AUTO
(8)  ---> OPTIONS DEVICE2=DASD DEVICE=TAPE
(9)  ---> CONTROL  DBNAME=dbname
(10) ---> BACKUP
          /*
```

Figure 71. Executing a Dual Backup in a VSE Environment

Statement 1 Specifies the DB2 and Data Restore feature libraries

Statement 2 Executes the procedure containing all the DB2 DLBLs.

Statement 3 Specifies ARCHIV as the *ddname* for the output file for the BACKUP function.

Statement 4 Assigns the file to a tape device. The output file is variable blocked with 32760-byte blocks.

Statement 5 Positions the tape at the beginning.

Statement 6 Specifies ARCHIV2 as the ddname of the second output file for the BACKUP function. Also note, **xxx** and **yyy** represent numerical values which should be large enough to hold the archive data.

Statement 7 Runs the program to execute the function.

Statement 8 Uses the OPTIONS statement to specify to take the second archive to DASD.

Statement 9 Specifies the name of the database in the control statement. This **dbname** is only used to name the output file.

Statement 10 Processes the BACKUP function.

The SYSLST report contains a list of all the processes executed. A function report produced by the BACKUP function is described in Figure 72 on page 100.

```
(1)  ---> XTS9-143 OPTIONS  DEVICE=TAPE  CONFIRM=NO
(1)  ---> XTS9-143 CONTROL DBNAME=dbname
(1)  ---> XTS9-143 BACKUP
          XTS9-143 /*
          XTS9-100 Data Restore feature VERSION 7.1.0
          XTS9-309 Processing DB2 Version 7 Release 3
(2)  ---> XTS9-172 DB2 was ended with LOGMODE L
(3)  ---> XTS9-141 External labeling of this archive is
          XTS9-142 Base SQL/VSE    date 17/10/95 - time 12:12:49
          XTS9-001 Processing directory
          XTS9-002           20010 Directory blocks saved
(4)  ---> XTS9-013 Table SQLDBA.COST_TABLE may be reloaded
          XTS9-013 Table SQLDBA.DEPARTMENT may be reloaded
          XTS9-013 Table SQLDBA.EMPLOYEE may be reloaded
          XTS9-013 Table SQLDBA.EMPLOYEE_ACTIVITY may be reloaded
          XTS9-013 Table SQLDBA.FOREIG may be reloaded
          XTS9-013 Table SQLDBA.INVENTORY may be reloaded
          XTS9-013 Table SQLDBA.OPERATIONS may be reloaded
          XTS9-013 Table SQLDBA.ORDERS may be reloaded
          XTS9-013 Table SQLDBA.PLAN_TABLE may be reloaded
          XTS9-013 Table SQLDBA.ROUTINE may be reloaded
          XTS9-145 Table SQLDBA.STORED QUERIES may be reloaded
          XTS9-013 Table SQLDBA.STRUCTURE_TABLE may be reloaded
          XTS9-013 Table SQLDBA.SUPPLIERS may be reloaded
          XTS9-013 Table SQLDBA.SYSLANGUAGE may be reloaded
          XTS9-013 Table SQLDBA.SYSTEXT1 may be reloaded
          XTS9-013 Table SQLDBA.SYSTEXT2 may be reloaded
          XTS9-013 Table SQLDBA.SYSUSERLIST may be reloaded
(5)  ---> XTS9-006 processing DDSK1
(5)  ---> XTS9-005              1 blocks saved
(5)  ---> XTS9-006 processing DDSK4
(5)  ---> XTS9-005              1 blocks saved
(6)  ---> XTS9-006 processing DDSK1
(7)  ---> XTS9-005           2870 blocks saved
          XTS9-006 processing DDSK2
          XTS9-005            383 blocks saved
          XTS9-006 processing DDSK3
          XTS9-005              7 blocks saved
          XTS9-006 processing DDSK4
          XTS9-005              1 blocks saved
          XTS9-006 processing DDSK5
          XTS9-005              1 blocks saved
          XTS9-006 processing DDSK6
          XTS9-005              1 blocks saved
          XTS9-006 processing DDSK7
          XTS9-005              1 blocks saved
          XTS9-006 processing DDSK8
          XTS9-005              1 blocks saved
          XTS9-007 processing successfully completed
```

Figure 72. Report after a Backup is Completed

Statement 1 Displays the contents of SYSIN on the SYSLST

Statement 2 Displays the log mode type when the server was ended.

Statement 3 Displays the external labeling for the BACKUP.

Statement 4 Displays the list of reloadable tables from the BACKUP file.

Statement 5 For some dbextents, header pages may be saved before other pages.

Statement 6 Displays each saved dbextent when processed.

Statement 7 Displays the number of dbextent blocks saved.

Step 3. Restarting the Database After Backing Up your Database

If you used the SQLEND UARCHIVE command to stop the application server, the next time you start the server, you are asked if the BACKUP process executed successfully as shown in Figure 73.

```
ARI0025I THE PROGRAM ARISQLDS IS LOADED AT 7F2078.
ARI0025I THE PROGRAM ARICMOD IS LOADED AT 905D80.
ARI0025I THE PROGRAM ARIXRDS IS LOADED AT 9CA000.
ARI0025I THE PROGRAM ARIXSXR IS LOADED AT B4A000
ARI0206D DID THE USER-ARCHIVE OF THE DIRECTORY AND 8
              DBEXTENT(S) EXECUTE SUCCESSFULLY
              REPLY 0(N0) OR 1(YES)
1
ARI0283I LOG ANALYSIS COMPLETE
ARI0282I LUW UNDO COMPLETE
ARI0281I LUW REDO COMPLETE
ARI0060I DATABASE MANAGER INITIALIZATION COMPLETE.
ARI0045I READY FOR OPERATOR COMMUNICATIONS
```

Figure 73. Restarting the Database After BACKUP When the Process Was Successful

When you answer **1** to indicate that the previous backup executed successfully, the history area is updated to specify that the BACKUP process completed without an error.

If an error occurred during the BACKUP function, answer **0** and the database server will restart without the BACKUP process completing successfully.

```
ARI0025I THE PROGRAM ARISQLDS IS LOADED AT 7F2078.
ARI0025I THE PROGRAM ARICMOD IS LOADED AT 905D80.
ARI0025I THE PROGRAM ARIXRDS IS LOADED AT 9CA000.
ARI0025I THE PROGRAM ARIXSXR IS LOADED AT B4A000
ARI0206D DID THE USER-ARCHIVE OF THE DIRECTORY AND 8
              DBEXTENT(S) EXECUTE SUCCESSFULLY
              REPLY 0(N0) OR 1(YES)
0
ARI0207D DO YOU WISH TO CONTINUE (WITHOUT USER-ARCHIVE) OR CANCEL
 (REDO THE USER-ARCHIVE)?
REPLY 'CONTINUE' OR 'CANCEL'
CONTINUE
ARI0283I LOG ANALYSIS COMPLETE
ARI0282I LUW UNDO COMPLETE
ARI0281I LUW REDO COMPLETE
ARI0060I DATABASE MANAGER INITIALIZATION COMPLETE.
ARI0045I READY FOR OPERATOR COMMUNICATIONS
```

Figure 74. Restarting the Database After BACKUP When the Process Was Not Successful

If you answered **0** to indicate that the BACKUP function did not complete successfully, you can choose to continue without the user archive or to redo the user archive. If you choose to continue without the user archive, the unsuccessful user archive is ignored. If you choose to redo the user archive, the CANCEL command stops the server and you must execute the BACKUP after the correcting the problem.

If you stop the application server using SQLEND when LOGMODE=Y, normal start up messages are displayed the next time the server is started as there is no record that you took a user archive.

Chapter 9. Restoring an Entire Database or an Entire Storage Pool

This chapter shows how to restore either an entire database or only damaged storage pools, when a database is damaged. If the database becomes corrupted, or one of the disks on which it resides is damaged, the database cannot be used. As a result, it is extremely important that the database administrator recover the lost data from a database archive, either by restoring a full database archive or by restoring specific storage pools.

The RESTORE function of Data Restore can restore databases from TRANSLATED DB2 Server for VSE & VM database archives and Data Restore archives. While you may want to do DB2 Server for VSE & VM database archives rather than Data Restore archives (for example, you may want to take a database archive without bringing down the database), the RESTORE function has the fastest performance when processing its own archive tapes. If you want to use the DB2 Server for VSE & VM database archives but also want to improve the performance of the RESTORE function, you must use the TRANSLATE function to convert the DB2 Server for VSE & VM database archives to the Data Restore archive format.

In performing this function, the DBA does the following steps:

1. Stops the application server, if it is still running.
2. Decides whether or not to restore from the last archive.
3. Determines whether a physical error is on the disk requiring a reformat of a dbextent.
4. Determines if it is necessary to restore the entire database or to restore only some storage pools
5. Starts the RESTORE function.
6. Determines whether a log recovery is also necessary.

Note: For pool recovery, the archive used as the base for pool recovery must include the current log in its restore set. If you are running in LOGMODE=A, only the most recent archive can be used. If you are running in LOGMODE=L, any archive may be used as long as the restore set is continuous and includes the current log. For example, if a COLDLOG is performed after the archive was taken, the COLDLOG breaks the continuity of the restore set for this archive, and so this archive can not be used for pool recovery.

Procedures to Recover from Failures

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.

   Note: If the input file is an incremental backup file, the incremental backup file will be processed first, then the FULL backup file will be required.
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.

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 the *DB2 Server for VSE & VM Diagnosis Guide and Reference* manual.

Data Recovery

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.

Note: If the input file is an incremental backup file, the incremental backup file will be processed first, then the FULL backup file will be required.

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, refer to "Log Recovery" on page 26.

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.

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.DB2vrm).

  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.DB2vrm).

  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.

Step 1. Stop the Application Server

Before you start to restore your database, you must stop your application server.

```
SQLEND
```

Figure 75. Stop the Application Server.

Step 2. Decide Whether to Restore from the Last Archive

If you intend to restore the database from the last user archive, go directly to Step 3.

If you want to use an archive other than the last one taken, you must identify the other archive to Data Restore.

For more information on the DESCRIBE function, see Chapter 13, "Displaying the Contents of an Archive File" on page 151.

Step 3. Determine If You Need to Reformat a DBEXTENT

For VM users: Before you execute the RESTORE function, you must decide if a minidisk needs redefining. If you need to redefine a minidisk, use the CMS FORMAT command and the RESERVE command. For details, refer to the section about replacing a minidisk in the *DB2 Server for VSE System Administration* or *DB2 Server for VM System Administration* manuals.

VSE users: Before you execute the RESTORE function, you must decide if the dbextent needs reformatting. For example, if the dbextent is damaged or has been redefined to a different location, it requires reformatting. If you need to reformat the dbextent, use a job similar to the one shown in Figure 76 in "Step 3A. Using the FORMAT Function (VSE Only)". Otherwise, you can start restoring your database as shown in "Step 4. Starting the RESTORE Function" on page 109.

Step 3A. Using the FORMAT Function (VSE Only)

When a disk is damaged or you move a dbextent to a different volume, redefine the dbextents using the IDCAMS DEFINE CLUSTER command. Before restoring the database, you must prepare the defined cluster for DB2. The FORMAT function does this process as shown in Figure 76.

Use the FORMAT command to format a VSE/VSAM defined file in the DB2 Server for VSE required format.

Using the FORMAT Command

Figure 76 shows an example of using the the FORMAT function.

```
         // JOB FORMAT
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // EXEC PROC=ARIS73DB
         // EXEC XTS91001,SIZE=AUTO
(3) ---> CONTROL DBNAME=dbname
(4) ---> FORMAT  DDSK=5
         /*
```

Figure 76. FORMAT Function Example

Statement 1 — Specifies the DB2 for VSE and Data Restore libraries for the FORMAT function.

Statement 2 — Contains all DLBL defining the directory, log, and all dbextents.

Statement 3 — Points to the appropriate database.

Statement 4	Formats the disk that was damaged.

Step 4. Starting the RESTORE Function

Before restoring, you need to decide if you want to restore the entire database or only the storage pools affected by a physical problem on disk.

First, determine which dbextents have been damaged, then execute the SHOWPOOL function to determine the storage pools associated with each dbextent (Refer to Chapter 15, "Displaying Pool Organization" on page 161 for more information on the SHOWPOOL command).

Storage pool level recovery cannot be used if any of the following conditions are met:

- if one of the pools is non-recoverable
- if you have more than 10 pools to restore
- if the log continuity has been broken (such as by a COLDLOG)
- if the database was running with LOGMODE=Y

If any of these conditions are met, then the entire database must be restored. Otherwise, only the affected storage pools need to be restored.

If you intend to use your last user archive to perform a full database RESTORE, use a job similar to the one shown in Figure 77 on page 110 for VSE or Figure 78 on page 110 for VM in the section "Step 4A. Restoring an Entire Database Using the Last Archive".

If you intend to use your last archive to perform SPLR, use a job similar to the one shown in Figure 80 on page 111 for VSE or Figure 81 on page 112 for VM in the section "Step 4B. Restoring Specific Storage Pools Using the Last Archive" on page 111.

If you do not intend to use your last user archive to perform a full database RESTORE, use a job similar to the one shown in Figure 91 on page 117 for VSE or Figure 92 on page 117 for VM in the section "Step 4E. Restoring Using INCREMENTAL Archive" on page 116.

If you do not intend to use your last archive to perform SPLR, use a job similar to the one shown in Figure 87 on page 115 for VSE or Figure 88 on page 115 for VM in the section "Step 4D. Restoring Specific Storage Pools Using Any Archive" on page 114.

Step 4A. Restoring an Entire Database Using the Last Archive

If you are in a VSE environment, restore your last archive using JCL similar to that in Figure 77 on page 110.

```
        // JOB RESTORE
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // EXEC PROC=ARIS73DB
(3) ---> // TLBL ARCHIV,'ARCHIVE.DB2',,,1
(4) ---> // ASSGN SYS006,180
(5) ---> // MTC REW,SYS006
(6) ---> // EXEC XTS91001,SIZE=AUTO
(7) ---> CONTROL DBNAME=dbname
(8) ---> RESTORE
(9) ---> /*
```

Figure 77. Sample of JCL (VSE) to Restore an Entire Database from the Last Archive

Statement 1 Specifies the DB2 for VSE for the RESTORE function.

Statement 2 Contains all DLBL defining the directory, log, and all dbextents.

Statement 3 Identifies the label on the tape.

Statement 4 Assigns a specific tape drive that will be used.

Statement 5 Rewinds the tape to its first file.

Statement 6 Executes the program XTS91001.

Statement 7 CONTROL statement specifying the **dbname** to control compatibility with BACKUP tape provided

Statement 8 Specifies the function to be performed. In this case, the RESTORE function.

Statement 9 Ends the SYSIN file.

If you are in a VM environment, restore your last archive using an EXEC similar to that in Figure 78.

```
        /**/
(1) ---> 'VMFPLC2 REW'
(2) ---> 'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
(3) ---> 'FILEDEF SYSPRINT DISK RESTORE SYSPRINT A'
(4) ---> 'FILEDEF SYSIN DISK RESTORE SYSIN A'
(5) ---> 'XTS91001'
```

Figure 78. Sample of JCL (VM) to Restore an Entire Database from the Last Archive

Statement 1 Rewinds the tape to its first file.

Statement 2 Identifies the file on the tape.

Statement 3 Specifies the destination of the SYSPRINT file.

Statement 4 Identifies the input source for the RESTORE function on SYSIN.

Statement 5 Executes the program XTS91001.

The SYSIN file must contain the following statements:

```
(6) ---> CONTROL DBNAME=dbname
(7) ---> RESTORE
```

Figure 79. Sample of SYSIN to Restore an Entire Database from the Last Archive

Statement 6 Points the user to the appropriate database.

Statement 7 Specifies the function to be performed; in this case, the RESTORE function.

Make sure that you can access the **dbname** SQLFDEF file on the database production disk when you use Data Restore.

Use the LINK command and the ACCESS command to access the Data Restore minidisk to execute the RESTORE function.

Step 4B. Restoring Specific Storage Pools Using the Last Archive

```
         // JOB RESTORE
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // EXEC PROC=ARIS73DB
(3) ---> // TLBL ARCHIV,'ARCHIVE.DB2',,,1
(4) ---> // DLBL DIRWORK,,,VSAM
(5) ---> // ASSGN SYS006,180
(6) ---> // MTC REW,SYS006
(7) ---> // EXEC XTS91001,SIZE=AUTO
(8) ---> CONTROL DBNAME=dbname
(9) ---> RESTORE POOL=(n, m, ...)
(10)---> /*
```

Figure 80. Sample of JCL (VSE) to Restore a List of Storage Pools from the Last Archive

Statement 1 Specifies the DB2 for VSE library for the RESTORE function.

Statement 2 Contains all DLBL defining the directory, log, and all dbextents.

Statement 3 Identifies the label on the tape.

Statement 4 Identifies the DIRWORK work file.

Statement 5 Assigns a specific tape drive that will be used.

Statement 6 Rewinds the tape to its first file.

Statement 7 Executes the program XTS91001.

Statement 8 CONTROL statement specifying the **dbname** to control compatibility with BACKUP tape provided

Statement 9 Specifies the function to be performed. In this case, the RESTORE function for some storage pools (maximum 10 pools).

Statement 10 Ends the SYSIN file.

If you are in a VM environment, restore your last archive using an EXEC similar to that in Figure 81 on page 112.

```
        /**/
(1) ---> 'VMFPLC2 REW'
(2) ---> 'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
(3) ---> 'FILEDEF DIRWORK DISK DIRWORK DATA H(RECFM F BLOCK 512'
(4) ---> 'FILEDEF SYSPRINT DISK RESTORE SYSPRINT A'
(5) ---> 'FILEDEF SYSIN DISK RESTORE SYSIN A'
(6) ---> 'XTS91001'
```

Figure 81. Sample of JCL (VM) to Restore a List of Storage Pools from the Last Archive

Statement 1 Rewinds the tape to its first file.

Statement 2 Identifies the file on the tape.

Statement 3 Identifies the DIRWORK work file.

Statement 4 Specifies the destination of the SYSPRINT file.

Statement 5 Identifies the input source for the RESTORE function on SYSIN.

Statement 6 Executes the program XTS91001.

The SYSIN file must contain the following statements:

```
(7) ---> CONTROL DBNAME=dbname
(8) ---> RESTORE POOL=(n, m, ...)
```

Figure 82. Sample of SYSIN to Restore a List of Storage Pools from the Last Archive

Statement 7 Points the user to the appropriate database.

Statement 8 Specifies the function to be performed; in this case, the RESTORE function for some storage pools (maximum 10 pools).

Make sure that you can access the **dbname** SQLFDEF file on the database production disk when you use Data Restore.

Use the LINK command and the ACCESS command to access the Data Restore minidisk to execute the RESTORE function.

At this point, you have restored the database or the storage pools.

Step 4C. Restoring an Entire Database Using Any Archive

If you are in a VSE environment, restore your archive using a job similar to that shows in Figure 83 on page 113.

```
        // JOB RESTORE
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // EXEC PROC=ARIS73DB
(3) ---> // TLBL ARCHIV,'ARCHIVE.DB2',,,1
(4) ---> // ASSGN SYS006,180
(5) ---> // MTC REW,SYS006
(6) ---> // EXEC XTS91001,SIZE=AUTO
(7) ---> CONTROL DATE=20/09/95 TIME=18:05:12, DBNAME=dbname
(8) ---> RESTORE
(9) ---> /*
```

Figure 83. Sample of JCL (VSE) to Restore an Entire Database from Any Archive

Statement 1 Specifies the DB2 Server for VSE & VM library for the RESTORE function.

Statement 2 Contains all the DLBL defining the directory, log, and all dbextents.

Statement 3 Identifies the label on the tape.

Statement 4 Assigns a specific tape drive that will be used.

Statement 5 Rewinds the tape to its first file.

Statement 6 Executes the program XTS91001.

Statement 7 Specifies instructions to the RESTORE function of Data Restore, for example, date and time. For more information about what date and time to specify, see Chapter 13, “Displaying the Contents of an Archive File” on page 151.

Statement 8 Specifies the function to be performed. In this case, the RESTORE function.

Statement 9 Ends the SYSIN file.

If you are in a VM environment, restore your archive using an EXEC similar to that in Figure 84.

```
        /**/
(1) ---> 'VMFPLC2 REW'
(2) ---> 'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
(3) ---> 'FILEDEF SYSPRINT DISK RESTORE SYSPRINT A'
(4) ---> 'FILEDEF SYSIN DISK RESTORE SYSIN A'
(5) ---> 'XTS91001'
```

Figure 84. Sample of Procedure (VM) to Restore an Entire Database from Any Archive

Statement 1 Rewinds the tape to its first file.

Statement 2 Identifies the file on the tape.

Statement 3 Specifies the destination of the SYSPRINT file.

Statement 4 Identifies the input source for the DESCRIBE function.

Statement 5 Executes the program XTS91001.

The SYSIN file must contain the following statements:

```
(6) ---> CONTROL DBNAME=dbname DATE=20/09/95 TIME=18:05:12
(7) ---> RESTORE
```

Figure 85. Sample of SYSIN to Restore an Entire Database from Any Archive

Statement 6 Specifies instructions for the RESTORE function of Data Restore; for example, date and time of the archives. For more information about what date and time to specify, see Chapter 13, "Displaying the Contents of an Archive File" on page 151.

Statement 7 Specifies the function to be performed; in this case, the RESTORE function.

Make sure that you can access the **dbname** SQLFDEF file on the database production disk when you use Data Restore.

Use the LINK command and the ACCESS command to access the Data Restore minidisk to execute the RESTORE function.

When you finish restoring the database, you see a list of all messages issued similar to the report shown in Figure 86.

```
XTS9-143 CONTROL DBNAME=dbname
XTS9-143 RESTORE
XTS9-143 /*
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-304 Restore from user archive invoked
XTS9-305 Current database will be destroyed
XTS9-196 Do you want to continue the RESTORE process ?
XTS9-406 Enter 0(Cancel) OR 1(Continue)
1
XTS9-136 Processing dbname archived on (17/10/95-12:12:49)
XTS9-010 Restoring DIRECTORY
XTS9-024           20010 Directory blocks restored
XTS9-011 Restoring DDSK1
XTS9-025            2870 blocks restored
XTS9-011 Restoring DDSK2
XTS9-025             383 blocks restored
XTS9-011 Restoring DDSK3
XTS9-025               7 blocks restored
XTS9-011 Restoring DDSK4
XTS9-025               1 block restored
XTS9-011 Restoring DDSK5
XTS9-025               1 block restored
XTS9-011 Restoring DDSK6
XTS9-025               1 block restored
XTS9-011 Restoring DDSK7
XTS9-025               1 block restored
XTS9-011 Restoring DDSK8
XTS9-025               1 block restored
XTS9-307 Startup the database manager with parameter "STARTUP=U"
XTS9-007 Processing successfully completed
```

Figure 86. Example of a Function Report After a Database Is Restored

Step 4D. Restoring Specific Storage Pools Using Any Archive

If you are in a VSE environment, restore your archive using a job similar to that shown in Figure 87 on page 115.

```
        // JOB RESTORE
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // EXEC PROC=ARIS73DB
(3) ---> // TLBL ARCHIV,'ARCHIVE.DB2',,,1
(4) ---> // DLBL DIRWORK,,,VSAM
(5) ---> // ASSGN SYS006,180
(6) ---> // MTC REW,SYS006
(7) ---> // EXEC XTS91001,SIZE=AUTO
(8) ---> CONTROL DATE=20/09/95 TIME=18:05:12, DBNAME=dbname
(9) ---> RESTORE POOL=(n, m, ...)
(10)---> /*
```

Figure 87. Sample of JCL (VSE) to Restore a List of Storage Pools from Any Archive

Statement 1 Specifies the DB2 for VSE library for the RESTORE function.

Statement 2 Contains all the DLBL defining the directory, log, and all dbextents.

Statement 3 Identifies the label on the tape.

Statement 4 Identifies the DIRWORK work file.

Statement 5 Assigns a specific tape drive that will be used.

Statement 6 Rewinds the tape to its first file.

Statement 7 Executes the program XTS91001.

Statement 8 Specifies instructions to the RESTORE function of Data Restore, for example, date and time. For more information about what date and time to specify, see Chapter 13, “Displaying the Contents of an Archive File” on page 151.

Statement 9 Specifies the function to be performed. In this case, the RESTORE function for some storage pools (maximum 10 pools).

Statement 10 Ends the SYSIN file.

If you are in a VM environment, restore your archive using an EXEC similar to that in Figure 88.

```
        /**/
(1) ---> 'VMFPLC2 REW'
(2) ---> 'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
(3) ---> 'FILEDEF DIRWORK DISK DIRWORK DATA H(RECFM F BLOCK 512'
(4) ---> 'FILEDEF SYSPRINT DISK RESTORE SYSPRINT A'
(5) ---> 'FILEDEF SYSIN DISK RESTORE SYSIN A'
(6) ---> 'XTS91001'
```

Figure 88. Sample of Procedure (VM) to Restore a List of Storage Pools from Any Archive

Statement 1 Rewinds the tape to its first file.

Statement 2 Identifies the file on the tape.

Statement 3 Identifies the DIRWORK work file.

Statement 4 Specifies the destination of the SYSPRINT file.

Statement 5 Identifies the input source for the DESCRIBE function.

Statement 6 Executes the program XTS91001.

The SYSIN file must contain the following statements:

```
(7) ---> CONTROL DBNAME=dbname DATE=20/09/95 TIME=18:05:12
(8) ---> RESTORE POOL=(n, m, ...)
```

Figure 89. Sample of SYSIN to Restore a List of Storage Pools from Any Archive

Statement 7 Specifies instructions for the RESTORE function of Data Restore; for example, date and time of the archives. For more information about what date and time to specify, see Chapter 13, "Displaying the Contents of an Archive File" on page 151.

Statement 8 Specifies the function to be performed; in this case, the RESTORE function.

Make sure that you can access the **dbname** SQLFDEF file on the database production disk when you use Data Restore.

Use the LINK command and the ACCESS command to access the Data Restore minidisk to execute the RESTORE function.

When you finish restoring some storage pools, you see a list of all messages issued similar to the report shown in Figure 90.

```
XTS9-143 CONTROL DBNAME=dbname
XTS9-143 RESTORE POOL=2
XTS9-143 /*
XTS9-196 Do you want to continue the RESTORE process ?
XTS9-406 Enter 0(Cancel) OR 1(Continue)
XTS9-403 Reply is 1
XTS9-136 Processing dbname archived on (11/06/96-17:06:37)
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
XTS9-406 Enter 0(Cancel) OR 1(Continue)
XTS9-403 Reply is 1
XTS9-006 Processing DDSK2
XTS9-010         383 blocks restored
XTS9-307 Startup the database manager with parameter "STARTUP=U"
XTS9-007 Processing successfully completed
```

Figure 90. Example of a Function Report After Some Storage Pools are Restored

Step 4E. Restoring Using INCREMENTAL Archive

The following examples are processing the restore of an entire database but the restore of specific storage pools can also be processed.

```
        // JOB RESTORE
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // EXEC PROC=ARIS73DB
(3) ---> // TLBL ARCHIV,'ARCHIVE.DB2',,,1
(4) ---> // TLBL FULLARC,'ARCHIVE.DB2',,,1
(5) ---> // ASSGN SYS006,180
(6) ---> // MTC REW,SYS006
(7) ---> // EXEC XTS91001,SIZE=AUTO
(8) ---> // OPTIONS DEVICE2=TAPE
(9) ---> CONTROL DATE=20/09/95 TIME=18:05:12, DBNAME=dbname
(10) ---> RESTORE
(11) ---> /*
```

Figure 91. Sample of JCL (VSE) to Restore an Entire Database from Any Archive

Statement 1 Specifies the DB2 Server for VSE & VM library for the RESTORE function.

Statement 2 Contains all the DLBL defining the directory, log, and all dbextents.

Statement 3 Identifies the label on the tape.

Statement 4 Identifies the label of the FULL archive.

Statement 5 Assigns a specific tape drive that will be used.

Statement 6 Rewinds the tape to its first file.

Statement 7 Executes the program XTS91001.

Statement 8 The OPTIONS statement specifies, if the FULLARC is on tape or DASD.

Statement 9 Specifies instructions to the RESTORE function of Data Restore, for example, date and time. For more information about what date and time to specify, see Chapter 13, “Displaying the Contents of an Archive File” on page 151.

Statement 10 Specifies the function to be performed. In this case, the RESTORE function.

Statement 11 Ends the SYSIN file.

If you are in a VM environment, restore your archive using an EXEC similar to that in Figure 92.

```
        /**/
(1) ---> 'VMFPLC2 REW'
(2) ---> 'FILEDEF FULLARC TAP1 SL 1 (RECFM VB BLOCK 3260'
(3) ---> 'FILEDEF SYSPRINT DISK RESTORE SYSPRINT A'
(4) ---> 'FILEDEF SYSIN DISK RESTORE SYSIN A'
(5) ---> 'XTS91001'
```

Figure 92. Sample of Procedure (VM) to Restore an Entire Database from Any Archive

Statement 1 Rewinds the tape to its first file.

Statement 2 Identifies the file on the tape.

Statement 3 Specifies the destination of the SYSPRINT file.

Statement 4 Identifies the input source for the DESCRIBE function.

Statement 5 Executes the program XTS91001.

The SYSIN file must contain the following statements:

```
(6) ---> CONTROL DBNAME=dbname DATE=20/09/95 TIME=18:05:12
(7) ---> RESTORE
```

Figure 93. Sample of SYSIN to Restore an Entire Database from Any Archive

Statement 6 Specifies instructions for the RESTORE function of Data Restore; for example, date and time of the archives. For more information about what date and time to specify, see Chapter 13, "Displaying the Contents of an Archive File" on page 151.

Statement 7 Specifies the function to be performed; in this case, the RESTORE function.

Make sure that you can access the **dbname** SQLFDEF file on the database production disk when you use Data Restore.

Use the LINK command and the ACCESS command to access the Data Restore minidisk to execute the RESTORE function.

When you finish restoring the database, you see a list of all messages issued similar to the report shown in Figure 94 on page 119.

```
XTS9-143 CONTROL DBNAME=dbname
XTS9-143 RESTORE
XTS9-143 /*
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-304 Restore from user archive invoked
XTS9-305 Current database will be destroyed
XTS9-196 Do you want to continue the RESTORE process ?
XTS9-406 Enter 0(Cancel) OR 1(Continue)
1
XTS9-136 Processing dbname archived on (17/10/95-12:12:49)
XTS9-010 Restoring DIRECTORY
XTS9-024           20010 Directory blocks restored
XTS9-011 Restoring DDSK1
XTS9-025            2870 blocks restored
XTS9-011 Restoring DDSK2
XTS9-025             383 blocks restored
XTS9-011 Restoring DDSK3
XTS9-025               7 blocks restored
XTS9-011 Restoring DDSK4
XTS9-025               1 block restored
XTS9-011 Restoring DDSK5
XTS9-025               1 block restored
XTS9-011 Restoring DDSK6
XTS9-025               1 block restored
XTS9-011 Restoring DDSK7
XTS9-025               1 block restored
XTS9-011 Restoring DDSK8
XTS9-025               1 block restored
XTS9-307 Startup the database manager with parameter "STARTUP=U"
XTS9-007 Processing successfully completed
```

Figure 94. Example of a Function Report After a Database Is Restored

Step 5. Deciding Whether to Use Log Recovery

At this point, you have restored the database or the storage pools.

Now, you need to make sure you maintain your data integrity. Restart the application server with the STARTUP=U parameter to indicate that you have restored the database from a user archive and want the log history area updated and the log recovery applied.

Figure 95 on page 120 shows an example of restarting after restoring the entire database.

```
ARI0025I THE PROGRAM ARISQLDS IS LOADED AT 7F2078.
ARI0025I THE PROGRAM ARICMOD IS LOADED AT 905D80.
ARI0025I THE PROGRAM ARIXRDS IS LOADED AT 9CA000.
ARI0025I THE PROGRAM ARIXSXR IS LOADED AT B1C000.
ARI0208D DID THE USER-RESTORE OF THE DIRECTORY AND 8
              DBEXTENT(S) EXECUTE SUCCESSFULLY
              ENTER 0(N0) OR 1(YES)
1
ARI0283I LOG ANALYSIS COMPLETE.
ARI0282I LUW UNDO COMPLETED.
ARI0281I LUW REDO IS COMPLETED.
ARI0060I DATABASE MANAGER INITIALIZATION COMPLETE.
ARI0045I READY FOR OPERATOR COMMUNICATIONS
```

Figure 95. Restart Your Database Manager After Restoring the Database

Note: If you restored some storage pools only but not the entire database, you must restart the database server with STARTUP=U parameter.

If a user archive (SQLEND UARCHIVE) was not used or you do not want to execute log recovery and you restored the entire database, do a COLDLOG to clear the contents of the current log.

Chapter 10. Backing Up Parts of a Database

In today's data processing environment, running global business operations in a 24 hours-a-day, 7 days-a-week mode is a key goal of many data centers. Data Restore helps data centers approach this goal by minimizing the time the database needs to be down.

It is for users who need the maximum database availability that the UNLOAD function is most useful. With the UNLOAD function, you can do a partial backup by unloading while the server is online or offline. If a physical error occurs, you can reload tables from the UNLOAD file after redefining the database. With the UNLOAD function, you can back up dbspaces containing critical tables more frequently than the rest of your database. If you have defined one table per dbspace, you effectively have table-level backup.

If a logical error occurs, you can reload all dbspaces in error from the unloaded tape or DASD.

Note: Forward log recovery is not available when restoring from unloaded dbspaces.

The primary purposes of the UNLOAD function is to allow you to:

- Unload dbspaces with the server online
- Unload dbspaces with the server offline
- Unload a single dbspace
- Unload multiple dbspaces

Note: Data Restore executes a physical unload of all pages containing data for the specified dbspaces. For that reason, the UNLOAD command and the RELOAD command of this feature and the UNLOAD command and the RELOAD command of the IBM DATABASE 2 Server for VSE & VM Services Utility (DBSU) are not compatible, so one cannot use the output of the other.

Using the UNLOAD Command

The UNLOAD command can include up to 90 dbspace names on a maximum of 10 SYSIN lines. In VSE, the REWIND parameter on the OPTIONS line, can be used to specify whether the output tape should be positioned (REWIND=YES) on not (REWIND=NO). In VM, you can use the LEAVE parameter on the FILEDEF for the output file to achieve the same results. Depending on the requirements, you can direct the UNLOAD function in a variety of ways as shown on page "UNLOAD" on page 186.

The UNLOAD command unloads dbspaces to an output file.

Unloading All Dbspaces

You can use an asterisk instead of a dbspace name to unload all the tables as shown in Figure 96 on page 122.

```
(1) ---> UNLOAD DBSPACE=(*)
```

Figure 96. Unload All Dbspaces in the Database

Statement 1 Indicates that all dbspaces are to be unloaded.

Figure 97 shows how to use the JCL to unload dbspaces in a VSE environment.

```
         // JOB UNLOAD
(1)  ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2)  ---> // EXEC PROC=ARIS73DB
(3)  ---> // TLBL ARCHIV,'UNLOAD.DB2'
(4)  ---> // ASSGN SYS006,180
(5)  ---> // MTC REW,SYS006
(6)  ---> // EXEC XTS91001,SIZE=AUTO, PARM='DBNAME(dbname)'
(7)  ---> OPTIONS DEVICE=TAPE,REWIND=YES/NO
(8)  ---> CONTROL DBAPW=XXXXXXXX,DBNAME=dbname
(9)  ---> UNLOAD  DBSPACE=(*)
         /*
```

Figure 97. JCL to Unload All Dbspaces

Statement 1 Specifies the DB2 library for the UNLOAD function.

Statement 2 Contains all DLBL defining directory, log, and all dbextents.

Statement 3 Identifies the label on the tape.

Statement 4 Assigns a specific tape drive that will be used.

Statement 5 Rewinds the tape to its first file.

Statement 6 Executes the program XTS91001.

Statement 7 Sets the device to a tape volume. In VSE, you can specify REWIND=YES or NO,(the default value is YES). If REWIND=NO is specified, at OPEN/CLOSE time the tape will not be positioned.

Statement 8 Uses the Control statement to specify the password for SQLDBA and identifies the **dbname** to be specified on the output file.

Statement 9 Specifies which database to UNLOAD. In this case, all dbspaces will be unloaded.

Figure 98 shows how to use an EXEC to unload dbspaces in a VM environment.

```
        /**/
(1) ---> 'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
(2) ---> 'FILEDEF SYSPRINT DISK UNLOAD  SYSPRINT A'
(3) ---> 'FILEDEF SYSIN DISK UNLOAD SYSIN A'
(4) ---> 'XTS91001'
```

Figure 98. EXEC to Unload All Dbspaces

The SYSIN file must contain the following statements:

```
(5) ---> CONTROL DBNAME=dbname DBAPW=XXXXXXXX
(6) ---> UNLOAD DBSPACE=(*)
```

Figure 99. SYSIN File to Specify Unloading All Dbspaces

Statement 1 Identifies the output file.

Statement 2 Identifies the SYSPRINT file

Statement 3 Identifies the SYSIN file.

Statement 4 Executes the program XTS91001.

Statement 5 Uses the Control statement to specify the password for the SQLDBA and identifies the **dbname** to be processed.

Statement 6 Specifies the type of UNLOAD to perform. In this case, an UNLOAD of all dbspaces will be performed.

Make sure that when you are executing Data Restore you can access the **dbname** SQLFDEF file on the database production disk.

Chapter 11. Restoring Logical Elements

You can reload tables from DB2 archive in one of two ways:

- Use the TRANSLATE function to convert the DB2 archive into a format that will minimize the time required for RELOAD. The TRANSLATE function can be run offline, after you have taken the DB2 archive, while the database is up.
- Run the RELOAD directly from the DB2 archive tape. The archive tape will be read twice during RELOAD processing.

There is a trade-off between running the TRANSLATE function to improve RELOAD processing and reading the DB2 archive tape twice if the TRANSLATE is not done.

This chapter contains all of the steps necessary to reload a table. You can reload directly from a Data Restore archive file, produced either by the BACKUP, BACKUP FULL/INCREMENTAL, or UNLOAD function.

Step 1 describes how to use the TRANSLATE function. You can use TRANSLATE to convert a DB2 archive into a format that improves RELOAD processing time.

Step 2 shows how to restore portions of the database from a BACKUP or UNLOAD file using the RELOAD function. Use RELOAD to recover the following:

- Existing tables after deleting existing rows
- New tables after creating the table
- Rows in an existing table
- Tables with all of their associated indexes, referential integrity, views, grants, comments, and labels
- Tables to prepare for forward recovery from log files

Step 3 shows how to list the changes recorded in the log archives and the current log for the tables being reloaded.

Step 4 shows how to apply the changes from the log archives and the current log.

Note: The reload process is done using DB2 commands, so this will create log records unless the RELOAD is run in single user mode with LOGMODE=N.

Recovery 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 "From a Physical Error" on page 104.

- Or restore the storage pool or the database as described in “From a Physical Error” on page 104. 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 “Recover from Directory Failure” on page 103.
- 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.

Step 1. Process a Standard DB2 Archive with Data Restore

Some databases must run 24 hours a day. Since user archives cannot be taken online, only DB2 database archives can be taken if the database must remain online.

There are two ways to process a DB2 archive tape to restore individual tables:

- If you rarely restore a table, you can specify the parameter ARCHTYPE=DB2 on an OPTIONS statement and proceed to step 2. Refer to “OPTIONS and CONTROL Statements” on page 167 for more information on the OPTIONS statement in the SYSIN file. Remember that the RELOAD function reads DB2 database archives twice but reads Data Restore archives only once.
- If you often restore tables from archives, or if you want to minimize the time necessary to restore the tables, execute the TRANSLATE function to convert the archive tapes produced by the ARCHIVE operator command into a Data Restore BACKUP file. This TRANSLATE function reads DB2 Database archive files twice, but produces an optimized file that can be used by the RELOAD function.

Notes:

1. You can only translate archive tapes produced by DB2 Version 7 Release 3
2. Take the online archive with the database in LOGMODE L.

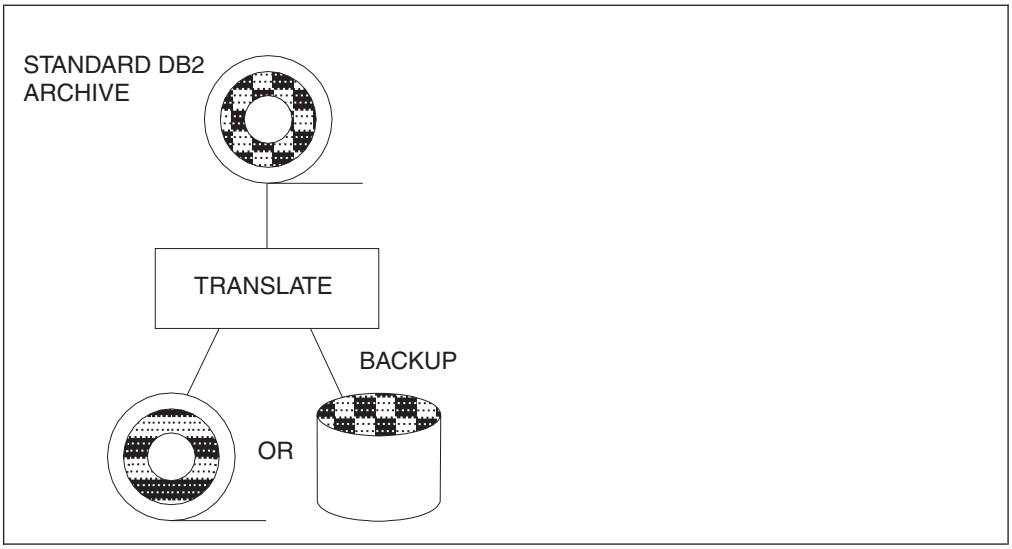


Figure 100. Translating a DB2 Server for VSE & VM Archive into a BACKUP

Before you can recover from DB2 Server for VSE & VM archives, you need to create some work files.

Recommended sizes for the work files are as follows:

1. For SYS0001, calculate the space required by using 4 KB for each active page in the DBSPACE.
2. For HEADER, use 4 KB for each HEADER page in the DBSPACE.
3. For DIRWORK, use the same size as the directory.
4. For archives, use the same size of the archive.

Procedures to Translate an Archive into a BACKUP File

Example of JCL (VSE) to translate an archive tape

```
        // JOB TRANSLATE
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // TLBL ARIARCH,'ARCHIVE.DB2'
(3) ---> // TLBL ARCHIV,'BACKUP.DB2'
(4) ---> // DLBL SYS0001,,,VSAM,RECSIZE=4096,RECORDS=(100,100)
(5) ---> // DLBL HEADER,,,VSAM,RECSIZE=4096,RECORDS=(100,100)
(6) ---> // DLBL DIRWORK,,,VSAM
(7) ---> // EXEC XTS91001,SIZE=AUTO
(8) ---> TRANSLATE
        /*
```

Figure 101. Sample JCL to Translate an DB2 Server for VSE Archive into a BACKUP File

Statement 1 Specifies the DB2 Server for VSE and Data Restore libraries.

Statement 2 Specifies the DB2 Server for VSE archive file with ddname ARIARCH.

Statement 3 Specifies the generated BACKUP file with ddname ARCHIV.

Statement 4 Specifies the work file SYS0001.

Statement 5 Specifies the work file HEADER.

Statement 6 Specifies the work file DIRWORK.

Statement 7 Runs the program XTS91001.

Statement 8 Specifies the TRANSLATE function.

Example of an EXEC to translate an archive tape

```
        /**/
(1) ---> 'FILEDEF ARIARCH TAP1 SL 1 (RECFM FB BLOCK 28672 LRECL 4096'
(2) ---> 'FILEDEF ARCHIV DISK BACKUP DATA G (RECFM VB BLOCK 32760'
(3) ---> 'FILEDEF SYS0001    DISK SYS0001   DATA H(RECFM F BLOCK 4096'
(4) ---> 'FILEDEF HEADER     DISK HEADER    DATA H(RECFM F BLOCK 4096'
(5) ---> 'FILEDEF DIRWORK DISK DIRWORK DATA H(RECFM F BLOCK 0512'
(6) ---> 'FILEDEF SYSIN DISK TRANS SYSIN    A'
(7) ---> 'FILEDEF SYSPRINT DISK TRANS SYSPRINT  A'
(8) ---> 'XEDIT TRANS SYSIN A'
(9) ---> 'XTS91001'
```

Figure 102. Sample EXEC to Translate a DB2 Server for VM Archive into a BACKUP File

Statement 1 Specifies the DB2 Server for VM archive file ddname ARIARCH.

Statement 2 Specifies the generated BACKUP file with ddname ARCHIV function.

Statement 3 Specifies the work file SYS0001.

Statement 4 Specifies the work file HEADER.

Statement 5 Specifies the work file DIRWORK.

Statement 6 Specifies the SYSIN file which contains the function specification.

Statement 7 Specifies the SYSPRINT file which contains the report.

Statement 8 Allows you to edit the SYSIN file to make any required changes before processing.

Statement 9 Runs the program XTS91001.

The SYSIN file must contain the following statements:

```
(1) ---> TRANSLATE
```

Figure 103. Sample SYSIN to Translate a DB2 Server for VM Archive into a BACKUP File

Statement 1 Specifies the TRANSLATE function.

Example of a Report from TRANSLATE

The list of all reloadable tables is displayed on SYSPRINT.

```
XTS9-143 OPTIONS
XTS9-143 TRANSLATE
XTS9-143 /*
XTS9-100 Data Restore feature VERSION 7.1.0
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   SQLDBA     .ALLGOOD                        MAY BE RELOADED
XTS9-013 Table   SQLDBA     .A0001                          MAY BE RELOADED
XTS9-013 Table   SQLDBA     .A0002                          MAY BE RELOADED
.
.
XTS9-013 Table   SQLDBA     .A0011                          MAY BE RELOADED
XTS9-007 Processing successfully completed
```

Figure 104. Sample Report after Translation of a DB2 Server for VM Archive

Notes:

1. The translate process reads DB2 Server for VM database archive tapes twice.
2. When the message XTS9-193 is displayed you have to remount the archive tapes beginning with the first archive tape.
3. You can restore the whole database from the translated DB2 Server for VM database archive, as well as restoring parts of the database.

Step 2. Reload Procedures

When a table has been corrupted by a program or accidentally dropped, you may have to restore that table. The following sections show the possibilities for reloading a table.

The database server must be active to reload tables, and applications using the server cannot access any table being reloaded during the reload process.

Note: The System Catalog tables cannot be reloaded.

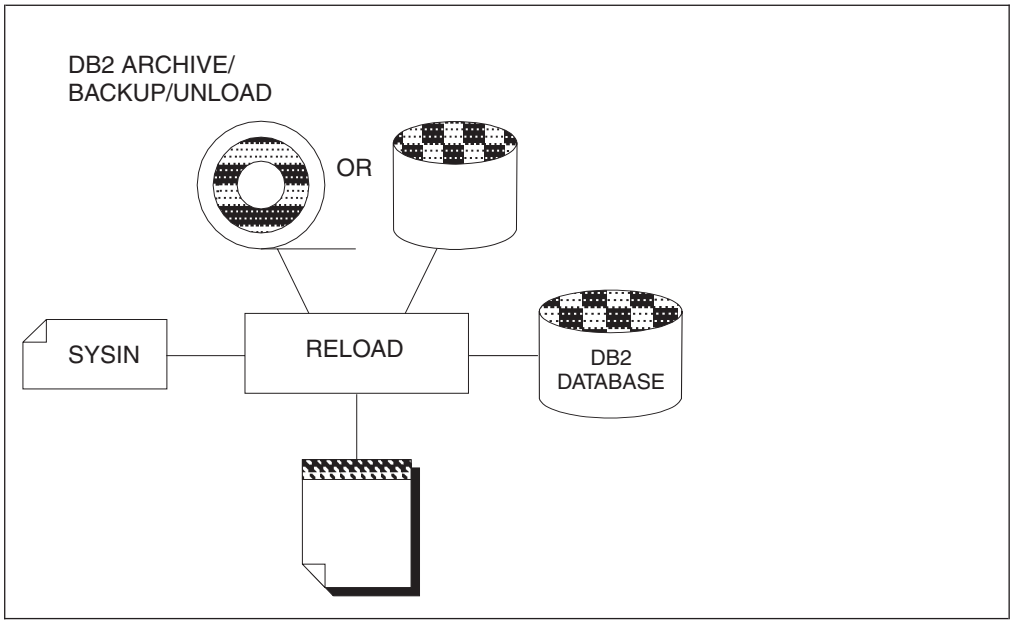


Figure 105. The RELOAD Function

Notes:

1. RELOAD can be used with BACKUP, UNLOAD, or DB2 Server for VSE & VM archive files.
2. The DB2 Server for VSE & VM database consists of the directory, all dbextents, and the log.
3. The SYSIN file will contain the list of tables to be restored with all the required parameters.

Using the RELOAD Command

To reload tables you use the RELOAD command.

The RELOAD function may also reload a table from one database to another. Specify the name of the source database on the CONTROL statement (parameter DBNAME) to verify that the file was produced from that database. If you are reloading directly from a DB2 Server for VSE & VM archive, specify ARCHTYPE=DB2 on the OPTIONS statement. If you are reloading from a translated DB2 Server for VSE & VM archive file, include the FILEDEFs or DLBLs for the DIRWORK, HEADER and SYS0001 files produced by the TRANSLATE function.

Reload a Table after Deleting Existing Rows

Sometimes you want to reload a table without keeping its current contents. To delete all of the rows in the table before starting the RELOAD process, specify PURGE for the FUNCT parameter.

You can execute this process from an UNLOAD or BACKUP file or from a DB2 Server for VSE & VM database archive file.

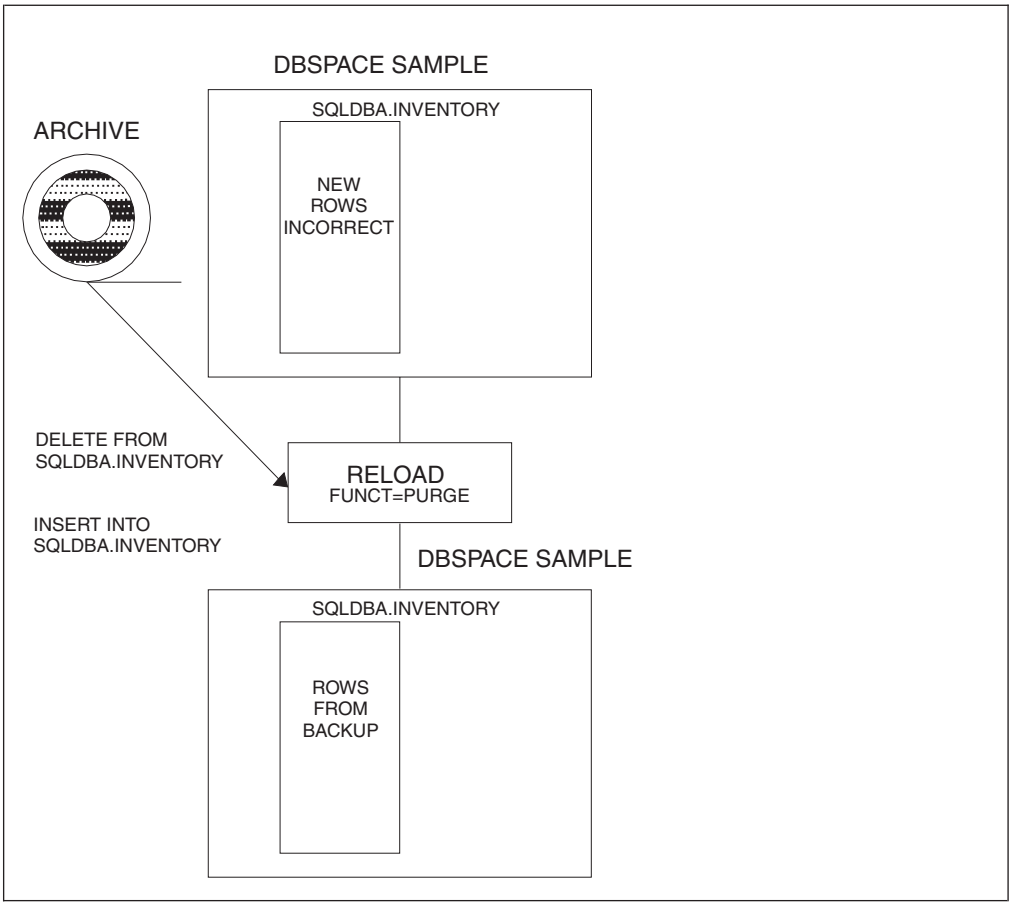


Figure 106. RELOADing a Table with the PURGE Parameter

Reload a New Table

Sometimes you want to reload a table that does not exist in the database (for example, a DROP table has been issued, or you are reloading into a different database) or you just want to reload into a table with a different name. Data Restore creates the table before reloading rows if you specify NEW for the FUNCT parameter and specify the DBSPACE where the table is to be created (using the DBSPACE parameter).

Note: For private dbspaces you must specify on owner (on the OWNER parameter).

You can execute this process from an UNLOAD or BACKUP file or from an DB2 Server for VSE & VM database archive file.

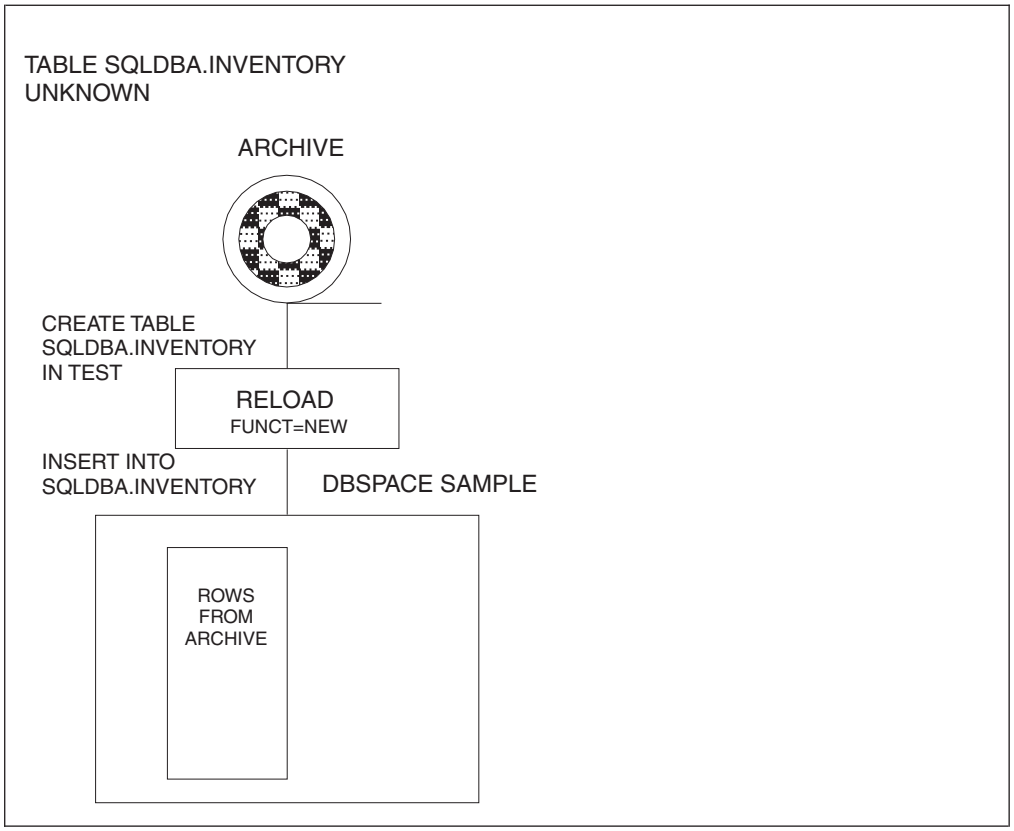


Figure 107. RELOADing a Table with the NEW Parameter

Reload Rows into an Existing Table

Sometimes you want to add new rows to an existing table from a BACKUP or UNLOAD file. To do this, specify ADD for the FUNCT parameter.

You can execute this process from an UNLOAD or BACKUP file or from an DB2 Server for VSE & VM database archive file.

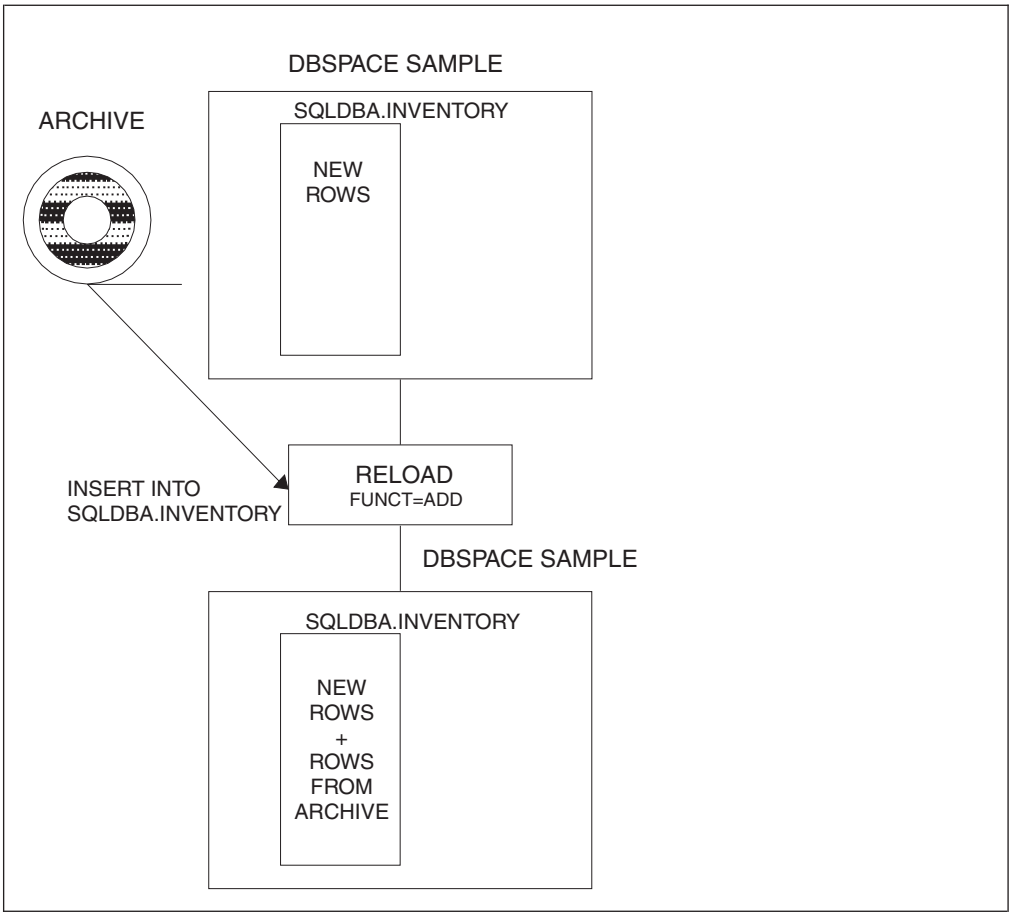


Figure 108. RELOADing a Table with the ADD Parameter

Reload a Table with Full Environment Recreation

Specify REPLACE for the FUNCT parameter to recreate the full environment on the table you want to reload.

If you specify REPLACE and the target table is the same as the source table (NEWTNAME and NEWCREATOR are not specified), the entire environment including indexes, referential integrity, views, grants, comments, and labels is recreated after the data are reloaded.

You can execute this process from an UNLOAD or BACKUP file or from an DB2 Server for VSE & VM database archive file.

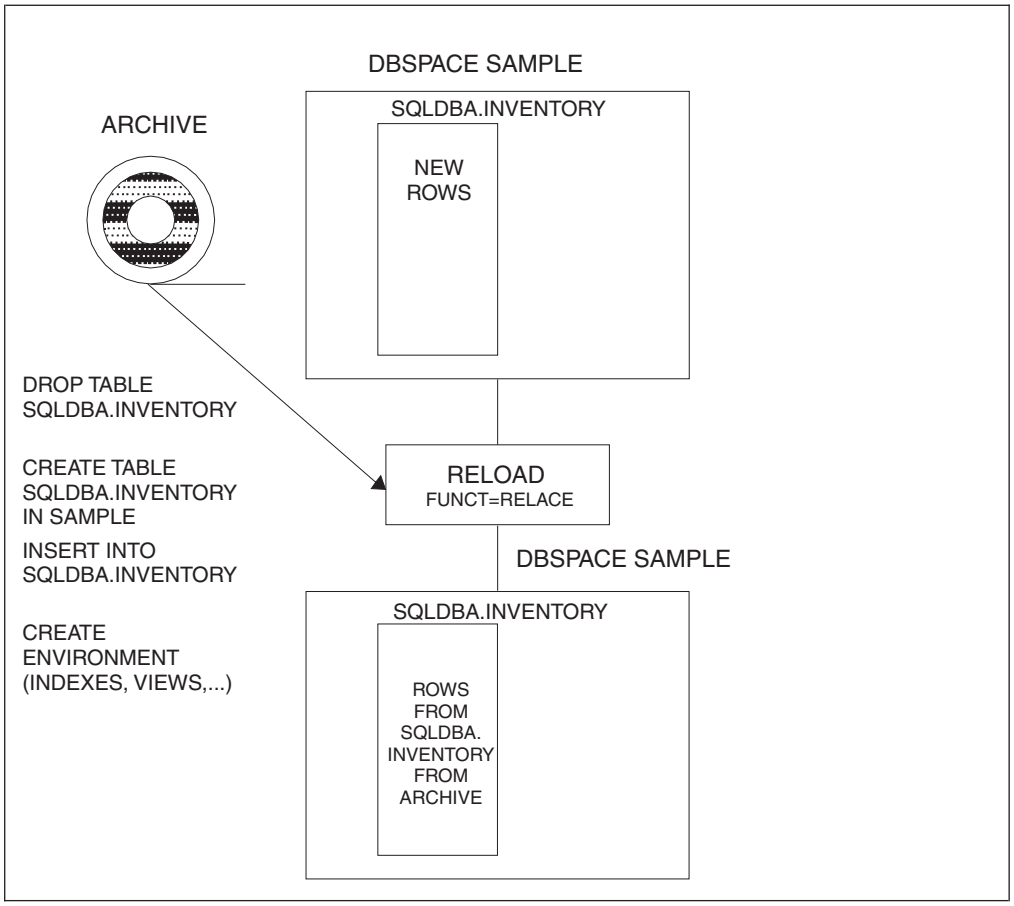


Figure 109. RELOADing a Table with the REPLACE Parameter

All of the commands for environment recreation will be displayed on SYSPRINT or SYSLST.

```
XTS9-143 CONTROL DBNAME=dbname  CONFIRM=NO
XTS9-143 OPTIONS COMMITCOUNT=600 CASE=M
XTS9-143 RELOAD  CREATOR=SQLDBA  TNAME=T1  FUNCT=REPLACE
XTS9-143 /*
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-136 Processing BASE1    archived on (11/10/94-15:19:11)
XTS9-127       600 rows loaded procedure  continuing
XTS9-127      1200 rows loaded procedure  continuing
XTS9-127      1800 rows loaded procedure  continuing
XTS9-127      2400 rows loaded procedure  continuing
XTS9-102      2650 rows loaded procedure  completed
XTS9-128         5 rows loaded into (SQLDBA.T1)
XTS9-128       933 rows loaded into (DATARFTR.SYSCOLUMNS)
XTS9-128       108 rows loaded into (DATARFTR.SYSCATALOG)
XTS9-128       693 rows loaded into (DATARFTR.SYSTABAUTH)
XTS9-128       106 rows loaded into (DATARFTR.SYSINDEXES)
XTS9-128        19 rows loaded into (DATARFTR.SYSVIEWS)
XTS9-128        14 rows loaded into (DATARFTR.SYSKEYCOLS)
XTS9-128        14 rows loaded into (DATARFTR.SYSKEYS)
XTS9-128       719 rows loaded into (DATARFTR.SYSUSAGE)
XTS9-128        39 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"."T1" ADD PRIMARY KEY ("C1" DESC) PCTFREE=10;
CREATE UNIQUE INDEX "SQLDBA"."IND1" ON "SQLDBA"."T1" ("C1" ASC ) PCTFREE=10;
ALTER TABLE "SQLDBA"."T1" ADD FOREIGN KEY "FOREIGNT2" ("C2")
REFERENCES "SQLDBA"."T2" ON DELETE RESTRICT;
COMMIT WORK ;
CONNECT SQLDBA  ;
GRANT SELECT ON "SQLDBA"."T1" TO "PAUL" WITH GRANT OPTION;
COMMIT WORK ;
CONNECT PAUL    ;
CREATE VIEW V1 AS SELECT * FROM SQLDBA.T1;
GRANT SELECT ON "PAUL"."V1" TO "GEORGE" WITH GRANT OPTION;
CREATE VIEW VV1 AS SELECT A.C1 FROM PAUL.V1 A,PAUL.V2 B WHERE  A.C1 = B.C2;
GRANT SELECT ON "PAUL"."VV1" TO "JOHN";
COMMIT WORK ;
CONNECT SQLDBA  ;
COMMENT ON TABLE "SQLDBA"."T1" IS 'TABLE T1';
COMMENT ON COLUMN "SQLDBA"."T1"."C1" IS 'COLUMN C1';
LABEL ON COLUMN "SQLDBA"."T1"."C1" IS '""';
COMMIT WORK ;
XTS9-007 Processing successfully completed
```

Figure 110. Example List Displayed for a RELOAD with the REPLACE Parameter

To recreate the environment, Data Restore creates copies of some of the System Catalog tables, using DATARFTR as the creator name; for example, information from SYSTEM.SYSCATALOG is loaded into DATARFTR.SYSCATALOG.

Reload Tables With Forward Recovery From the Log

When you reload a table from a BACKUP or a DB2 Server for VSE & VM archive file, the reloaded table does not contain any of the changes made on that table after the backup was taken. Those changes are recorded in any log archives taken since the backup and in the current log.

If you want to reload a table and apply the changes, specify RECOVERY=YES on the OPTIONS statement (refer to "OPTIONS and CONTROL Statements" on page 167) during the RELOAD process. Refer to the next sections for information on listing and applying the changes in the log.

Notes:

1. This function is available for BACKUP or DB2 Server for VSE & VM archive files processed when the server was running in LOGMODE A or L.
2. Forward Recovery can only be applied to tables that exist in the archive that is being used for the RELOAD.
3. Forward log recovery stops when it encounters a DROP TABLE or DROP DBSPACE command in the log.
4. If dual logging is being used, Data Restore will switch to the secondary log if there is an error processing the primary log.
5. Alternate Logging is fully supported by Data Restore and no modification is needed.

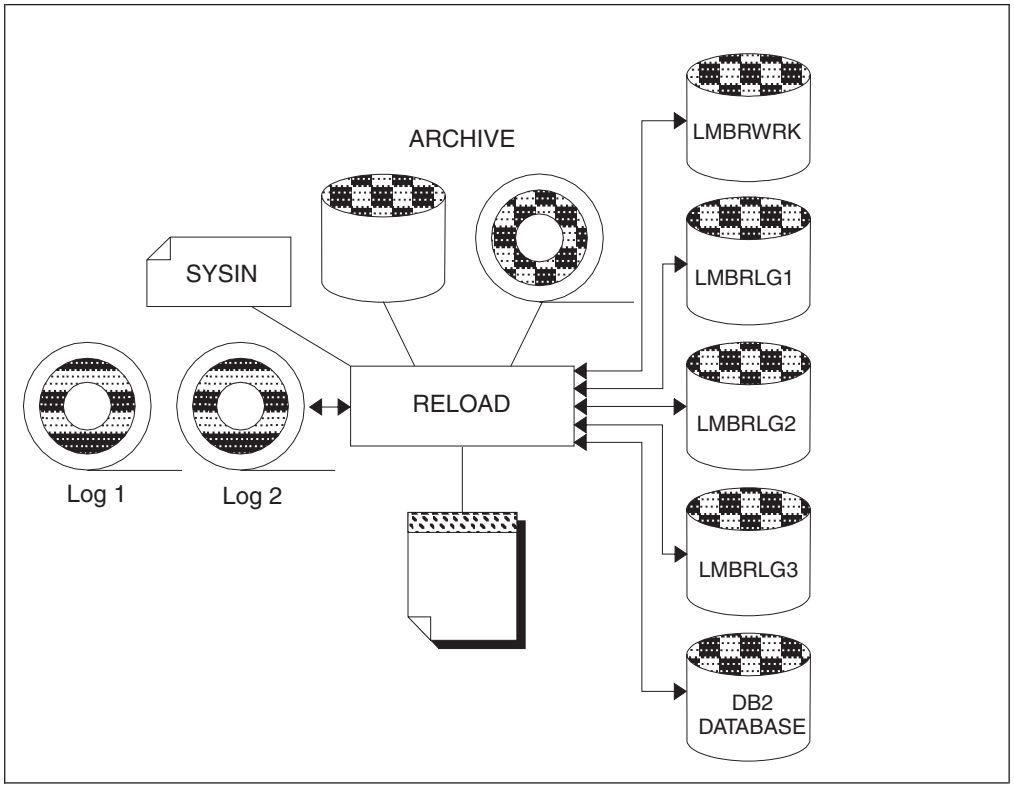


Figure 111. Possible Input Sources for RELOADing Tables with Forward Recovery

```
       // JOB RELOAD
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL LMBRWRK,,,VSAM
(3) ---> // TLBL ARCHIV,'ARCHIVE.SQL',,,1
(4) ---> // ASSGN SYS006,180
(5) ---> // MTC REW,SYS006
(5) ---> // EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(DBNAME)'
(7) ---> CONTROL DBNAME=DBNAME
(7) ---> OPTIONS DEVICE=TAPE
(7) ---> RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
(7) ---> RELOAD CREATOR=SQLDBA TNAME=SUPPLIERS
(7) --->        FUNCT=NEW
(7) --->        NEWTNAME=NEWSUPPLIERS
(7) --->        DBSPACE=SAMPLE
       /*
```

Figure 112. Example of JCL to Reload in a VSE Environment From a Full Backup

Statement 1 DB2 library must be specified for the RELOAD function.

Statement 2 LMBRWRK is used if the reloaded tables contain LONG columns.

Statement 3 ARCHIV is the ddname for the BACKUP file. It can be a BACKUP file or a FULL BACKUP.

Statement 4 The tape containing the backup file is assigned.

Statement 5 The tape is rewound.

Statement 6 Program to execute functions is called. The DBNAME of the database to process must be specified on the PARM option.

Statement 7 SYSIN for the program is specified.

```
         // JOB RELOAD
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL LMBRWRK,,,VSAM
(3) ---> // TLBL ARCHIV,'ARCHIVE.SQL',,,1
(4) ---> // TLBL FULLARC,'ARCHIVE.SQL',,,1
(5) ---> // ASSGN SYS006,180
(6) ---> // MTC REW,SYS006
(7) ---> // EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(dbname)'
(8) ---> CONTROL DBNAME=dbname WRKSIZE=4096
(8) ---> OPTIONS DEVICE=TAPE DEVICE2=TAPE
(8) ---> RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
(8) ---> RELOAD CREATOR=SQLDBA TNAME=SUPPLIERS
(8) --->        FUNCT=NEW
(8) --->        NEWTNAME=NEWSUPPLIERS
(8) --->        DBSPACE=SAMPLE
        /*
```

Figure 113. Example of JCL to Reload in a VSE Environment From an Incremental Backup

Statement 1 DB2 library must be specified for the RELOAD function.

Statement 2 LMBRWRK is used if the reloaded tables contain LONG columns.

Statement 3 ARCHIV is the ddname for the last Incremental Backup.

Statement 4 FULLARC is the ddname for the last Full Backup.

Statement 5 The tape containing the backup file is assigned.

Statement 6 The tape is rewound.

Statement 7 Program to execute functions is called. The DBNAME of the database to process must be specified on the PARM option.

Statement 8 SYSIN for the program is specified. WRKSIZE must be equivalent to the parameter specified during the BACKUP function. DEVICE2 specifies if a TAPE or DASD (TLBL/DLBL) is used for Full Backup (FULLARC).

```
         // JOB RELOAD
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL LMBRWRK,,,VSAM
(3) ---> // DLBL LMBRLG1,,,VSAM
(3) ---> // DLBL LMBRLG2,,,VSAM
(3) ---> // DLBL LMBRLG3,,,VSAM
(4) ---> // DLBL ARCHIV,,,VSAM
(5) ---> // TLBL LARCHIV,,,,1
(6) ---> // TLBL ARIARCH,'ARCHIVE.SQL',,,1
(7) ---> // ASSGN  SYS006, 180
(8) ---> // MTC REW,SYS006
(9) ---> // DLBL SYS0001,,,VSAM
(9) ---> // DLBL HEADER,,,VSAM
(9) ---> // DLBL DIRWORK,,,VSAM
(10) --->  // EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(dbname)'
(11) --->  OPTIONS  ARCHTYPE=SQLDS
(11) --->  CONTROL  DBNAME=dbname
(11) --->  RELOAD  CREATOR=SQLDBA,TNAME=CUSTOMERS FUNCT=NEW
(11) --->  DBSPACE=SAMPLE
```

Figure 114. Example of JCL to Reload in a VSE Environment From a DB2 Archive

Statement 1 DB2 library must be specified for the RELOAD function.

Statement 2 LMBRWRK is used if the reloaded tables contain LONG columns.

Statement 3 LMBRLG1, LMBRLG2, and LBMRLG3 are used to extract all referenced in log files for the reloaded tables when RECOVERY=YES parameter is specified. They will be used to execute forward log recovery.

Statement 4 ARCHIV is the work file for this process.

Statement 5 LARCHIV is the ddname for the Log archive files.

Statement 6 ARIARCH is the ddname for the DB2 database archive file.

Statement 7 The tape containing the archive file is assigned.

Statement 8 The tape is rewound.

Statement 9 SYS0001, HEADER, DIRWORD are work files for the function.

Statement 10 Program to execute functions is called. The DBNAME of the DBNAME of the database to process must be specified on PARM option.

Statement 11 SYSIN for the program is specified. OPTIONS statement specifies that process is executed from DB2 database archive.

```
/**/
'TAPE REW'
'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
'FILEDEF LMBRWRK DISK LMBRWRK DATA A (RECFM FB BLOCK 28672 LRECL 4096'
'FILEDEF SYSPRINT DISK RELOAD SYSPRINT A'
'FILEDEF SYSIN DISK RELOAD SYSIN A'
'XEDIT RELOAD SYSIN A'
'XTS91001'


 SYSIN file must contain the following statements :

 CONTROL DBNAME=dbname
 RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
 RELOAD CREATOR=SQLDBA TNAME=SUPPLIERS
    FUNCT=NEW
    NEWTNAME=NEWSUPPLIERS
```

Figure 115. Example of VM Procedure to Reload From a Backup File

```
/**/
'TAPE REW'
'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
'FILEDEF FULLARC TAP1 SL1 (RECFM VB BLOCK 32760'
'FILEDEF LMBRWRK DISK LMBRWRK DATA A (RECFM FB BLOCK 28672 LRECL 4096'
'FILEDEF SYSPRINT DISK RELOAD SYSPRINT A'
'FILEDEF SYSIN DISK RELOAD SYSIN A'
'XEDIT RELOAD SYSIN A'
'XTS91001'

SYSIN file must contain the following statements :

OPTIONS WRKSIZE=4096 DEVICE2=TAPE
CONTROL DBNAME=dbname
RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
RELOAD CREATOR=SQLDBA TNAME=SUPPLIERS
   FUNCT=NEW
   NEWTNAME=NEWSUPPLIERS
   DBSPACE=SAMPLE
```

Figure 116. Example of VM Procedure to Reload From an Incremental Backup

```
       /**/
(1) ---> 'FI LMBRLG1 DISK LMBRLG1 DATA A (RECFM VB BLOCK 32760'
        'FI LMBRLG2 DISK LMBRLG2 DATA A (RECFM VB BLOCK 32760'
        'FI LMBRLG3 DISK LMBRLG3 DATA A (RECFM VB BLOCK 32760'
(2) ---> 'FI ARIARCH TAP1 SL (RECFM FB BLOCK 28672 LRECL 4096'
(3) ---> 'FI ARCHIV DISK ARCHIV DATA  G (RECFM VB BLOCK 32760'
(4) ---> 'FI LARCHIV DISK BASE2 LOGDSK1  A1(RECFM F BLOCK 4096'
(5) ---> 'FI LMBRWRK DISK LMBRWRK DATA H (RECFM FB BLOCK 28672
         LRECL 4096'
(6) ---> 'FI SYS0001 DISK SYS0001 DATA H(RECFM F BLOCK 4096'
(7) ---> 'FI HEADER  DISK HEADER  DATA H(RECFM F BLOCK 4096'
(8) ---> 'FI DIRWORK DISK DIRWORK DATA H(RECFM F BLOCK 0512'
(9) ---> 'FI SYSIN DISK RELOAD   SYSIN  A'
        'FI SYSPRINT DISK RELOAD   SYSPRINT A'
        'XEDIT RELOAD SYSIN A'
        'XTS91001'                          /*Call BACKUP/RESTORE*/
```

Figure 117. Example of VM Procedure to Reload From a DB2 Archive

Statement 1 LMBRLG1, LMBRLG2, LMBRLG3 files are used for recovery processing.

Statement 2 Defines the virtual tape drive for SQL/DS archive file.

Statement 3 Defines a work file. This file will contain on SQLDA per reloadable table.

Statement 4 Defines the database log file.

Statement 5 Defines the work file to process function.

Statement 6 Defines the file to contain all pages for dbspace SYS0001

Statement 7 Defines the file to contain all header pages.

Statement 8 Defines the files to contain all directory pages.

Statement 9 Defines the SYSIN file.

Step 3. List the Changes Extracted from the Log Files

The RELOAD function can reload tables from a BACKUP file. All changes executed on the tables after the BACKUP function are recorded in the log.

After a RELOAD function, if the parameter RECOVERY=YES was specified on the OPTIONS statement (for more information see "OPTIONS Statement Parameters" on page 168; the LMBRLG1, LMBRLG2, LMBRLG3 files contain all the SQL statements from the log that affect the table.

You can use the LISTLOG function to list the contents of these files. The APPLYLOG function must be used to apply the changes in these files. Tables reloaded from an UNLOAD file cannot have log changes applied.

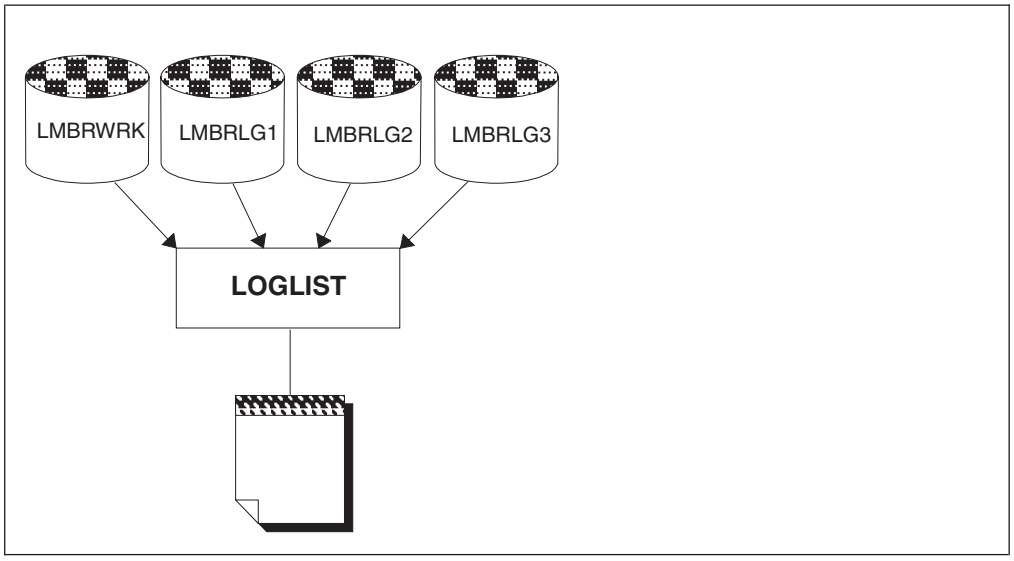


Figure 118. List the Changes Extracted from the Log Input work files:

When You Need to Execute the LISTLOG Function

When you reload a table and request the RECOVERY process, all changes referenced in the log will be extracted. If, for example, you are recovering a table that was accidentally corrupted, and you reapply all the changes, the LUW that

corrupted the table will also be reexecuted. You need to know the timestamp of that LUW so that you can reapply all changes up to but not including the LUW in error.

The timestamp is used during the APPLYLOG function (refer to "Step 4. Apply LUWs Referenced in the Log to the Reloaded Tables" on page 143 for more information) on the END parameter to apply the log changes up to the LUW in error.

Note: The timestamp of each LUW and SQL command has the following format:
`YYYY-DDD-HH-MM-SS-mmmmmm`

Example: 1995-260-23-59-59-000000

Using the LISTLOG Command

To list the contents of LMBRLG1, LMBRLG2, and LMBRLG3, execute the LISTLOG function using the XTS91001 program.

The following is an example of JCL (VSE) to list the contents of the recovery files.

```
         // JOB LISTLOG
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL LMBRWRK,,,VSAM
(3) ---> // DLBL LMBRLG1,,,VSAM
(3) ---> // DLBL LMBRLG2,,,VSAM
(3) ---> // DLBL LMBRLG3,,,VSAM
(4) ---> // EXEC XTS91001,SIZE=AUTO
(5) ---> CONTROL DBAPW=XXXXXXXX
(6) ---> LISTLOG
         /*
```

Figure 119. Example JCL to List the Contents of the Recovery Files

Statement 1 Specifies the DB2 Server for VSE and Data Restore libraries.

Statement 2 Specifies the work file LMBRWRK which may be used if the table contains LONG columns.

Statement 3 Specifies the work files LMBRLG1, LMBRLG2, and LMBRLG3 which contain data extracted from the log during the RELOAD process.

Statement 4 Runs the program XTS91001.

Statement 5 Specifies the password for user SQLDBA.

Statement 6 Specifies the LISTLOG function.

The following is an example of an EXEC to list the contents of the recovery files.

```
        /**/
(1) ---> 'FILEDEF SYSIN DISK LISTLOG SYSIN A'
(2) ---> 'FILEDEF SYSPRINT DISK LISTLOG SYSPRINT A'
(3) ---> 'FILEDEF LMBRWRK DISK LMBRWRK DATA A (RECFM FB BLOCK 28672 LRECL 4096'
(4) ---> 'FILEDEF LMBRLG1 DISK LMBRLG1 DATA A (RECFM VB BLOCK 32760'
(4) ---> 'FILEDEF LMBRLG2 DISK LMBRLG2 DATA A (RECFM VB BLOCK 32760'
(4) ---> 'FILEDEF LMBRLG3 DISK LMBRLG3 DATA A (RECFM VB BLOCK 32760'
(5) ---> 'XTS91001'
```

Figure 120. Sample Procedure to List the Content of the Recovery Files

Statement 1 Specifies the SYSIN file which contains the function specification.

Statement 2 Specifies the SYSPRINT file which contains the function report.

Statement 3 Specifies the work file LMBRWRK which is used if the table contains LONG columns.

Statement 4 Specifies the work files LMBRLG1, LMBRLG2, and LMBRLG3 which contain data extracted from the log during the RELOAD process.

Statement 5 Runs the program XTS91001.

The SYSIN file must contain the following statements:

```
(1) ---> CONTROL DBAPW=XXXXXXXX
(2) ---> LISTLOG
```

Figure 121. Control Statement for LISTLOG in a VM Environment

Statement 1 Specifies the password for user SQLDBA.

Statement 2 Specifies the LISTLOG function.

Example of a Report from LISTLOG Execution

For each row in the LMBRLG1 file, two lines are displayed:

1. An identification line that contains three pieces of information: a **committed** or **rolled back** indicator (**C** or **R**), the LUWID, and the timestamp.
2. The SQL statement that was executed.

```
(1) ---> C 00004084 1995-251-10-56-29-861568
(2) ---> INSERT INTO "SQLDBA"."CUSTOMERS" VALUES ('TEST',0,200)
(3) ---> R 00004092 1995-256-10-58-00-126994
(4) ---> DELETE FROM "SQLDBA"."CUSTOMERS" WHERE "CUST_NO"=1245
```

Figure 122. Example Report After LISTLOG Function Execution

Statement 1 Specifies that the LUW number 00004084 was committed at 10-56-29861568 on day 251 in 1995.

Statement 2 Shows that the SQL statement executed was an INSERT into the table SQLDBA.CUSTOMERS.

Statement 3 Specifies that LUW number 00004092 was rolled back at 10-58-00-126994 on day 251 in 1995.

Statement 4 Shows that the SQL statement executed was a DELETE from table SQLDBA.CUSTOMERS.

Step 4. Apply LUWs Referenced in the Log to the Reloaded Tables

After a RELOAD function has been processed, the table contains rows as they appeared in the table when the BACKUP was taken. All changes executed on the table after the BACKUP function was taken can be reprocessed by Data Restore on the reloaded table.

If among the log changes on a table, a DELETE FROM TABLE command was issued in error, you may want to stop the recovery before the DELETE command is reexecuted. By specifying END=**timestamp**, the recovery process stops before the timestamp. If a DROP command that involves one of the reloaded tables is encountered in the log, the recovery process stops automatically for the table.

Forward log recovery can only be processed from a BACKUP archive (not an UNLOAD) and on the original database. You cannot do log recovery after reloading a table into a different database.

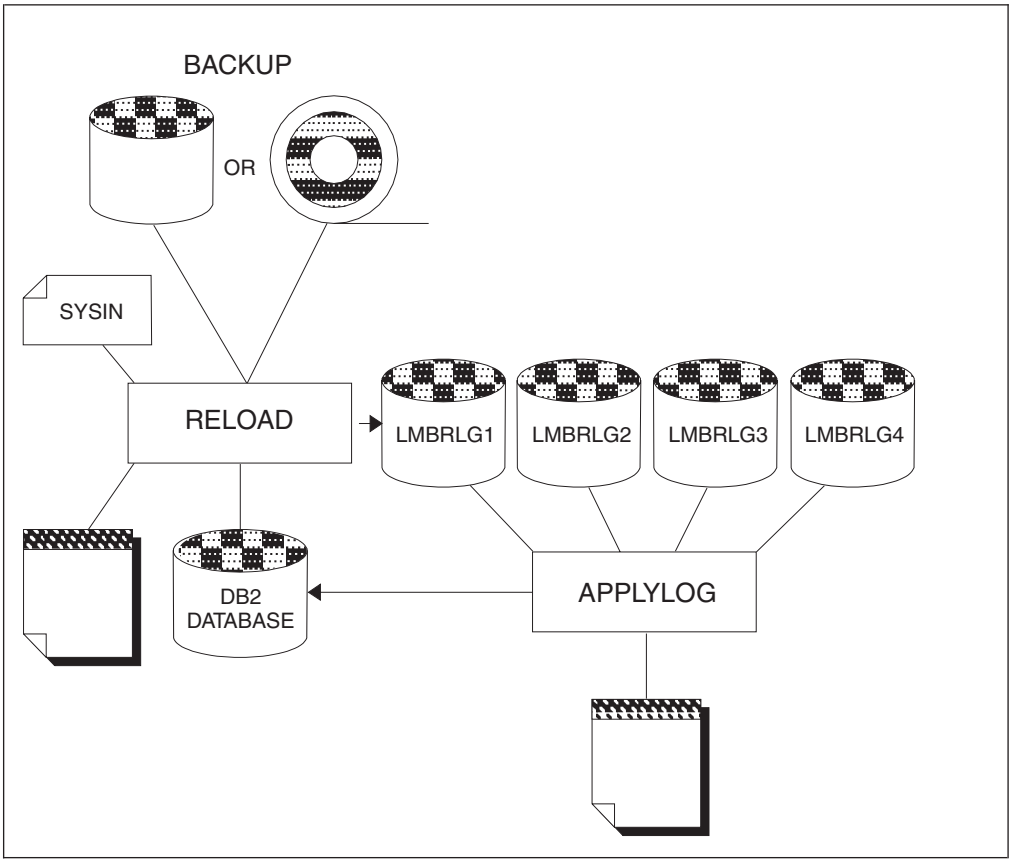


Figure 123. Apply Changes From the Log To the Reloaded Tables

Using the APPLYLOG Command

To determine the value to specify for the END parameter, execute the LISTLOG function (for more information refer to "Step 3. List the Changes Extracted from the Log Files" on page 140). An example of the use of TIMESTAMP is: TIMESTAMP=1995-320-12-30-01-000000.

Notes:

1. All LUWs whose timestamp is equal to or greater than the specified END timestamp will not be executed.
2. The database server must be active during the APPLYLOG function and all changes on the database are logged.
3. If you want to apply all changes, specify a timestamp greater than the timestamp of the last LUW in the log.

Execute the APPLYLOG Function

To process all the SQL statements in LMBRLG1, LMBRLG2, and LMBRLG3, execute the APPLYLOG function using program XTS91001.

You must have previously run the RELOAD function, from a BACKUP, with the RECOVERY=YES parameter to reload the tables.

The following is an example of JCL (VSE) to execute the statements in the recovery files.

```
         // JOB APPLYLOG
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL LMBRWRK,,,VSAM
(3) ---> // DLBL LMBRLG1,,,VSAM
(3) ---> // DLBL LMBRLG2,,,VSAM
(3) ---> // DLBL LMBRLG3,,,VSAM
(4) ---> // EXEC XTS91001,SIZE=AUTO
(5) ---> CONTROL DBAPW=XXXXXXXX
(6) ---> APPLYLOG END=1995-260-23-59-59-000000
         /*
```

Figure 124. Sample JCL to Execute SQL Statements in the Recovery Files

Statement 1 Specifies the DB2 Server for VSE Data Restore libraries.

Statement 2 Specifies the work file LMBRWRK which is used if the table contains LONG columns.

Statement 3 Specifies the work files LMBRLG1, LMBRLG2, and LMBRLG3 which contain data extracted from the log during the RELOAD process.

Statement 4 Runs the program XTS91001.

Statement 5 Specifies the password for user SQLDBA.

Statement 6 Specifies the APPLYLOG function with the END timestamp.

The following is an example on an EXEC to execute the SQL statements in the recovery files.

```
        /**/
(1) ---> 'FILEDEF SYSIN DISK LISTLOG SYSIN A'
(2) ---> 'FILEDEF SYSPRINT DISK LISTLOG SYSPRINT A'
(3) ---> 'FILEDEF LMBRWRK DISK LMBRWRK DATA A (RECFM FB BLOCK 28672 LRECL 4096'
(4) ---> 'FILEDEF LMBRLG1 DISK LMBRLG1 DATA A (RECFM VB BLOCK 32760'
(4) ---> 'FILEDEF LMBRLG2 DISK LMBRLG2 DATA A (RECFM VB BLOCK 32760'
(4) ---> 'FILEDEF LMBRLG3 DISK LMBRLG3 DATA A (RECFM VB BLOCK 32760'
(5) ---> 'XTS91001'
```

Figure 125. Sample EXEC to Execute SQL Statements in the Recovery Files

Statement 1 Specifies the SYSIN file which contains the function specification.

Statement 2 Specifies the SYSPRINT file which contains the output report.

Statement 3 Specifies the work file LMBRWRK which is used if the table contains LONG columns.

Statement 4 Specifies the work files LMBRLG1, LMBRLG2, and LMBRLG3 which contain data extracted from the log during the RELOAD process.

Statement 5 Runs the program XTS91001.

The SYSIN file must contain the following statements:

```
(1) ---> CONTROL DBAPW=XXXXXXXX
(2) ---> APPLYLOG END=1995-260-23-59-59-000000
```

Figure 126. Sample SYSIN for the APPLYLOG Function

Statement 1 Specifies the password for user SQLDBA.

Statement 2 Specifies the APPLYLOG function with the END timestamp.

Example of a Report from APPLYLOG

```
XTS9-143 OPTIONS      LANG=S001
XTS9-143 CONTROL DBAPW=********
XTS9-143 APPLYLOG  END=1996-255-16-46-47-646175
XTS9-143 /*
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-007 Processing successfully completed
```

Figure 127. Example of a Report from APPLYLOG Execution

Chapter 12. Accessing Data Whether the Server is Up or Down

Introduction

With the SELECT function, you can extract data from the database directly from the dbextents. This function is available regardless of whether the database server is online or offline. If SELECT is used when the database server is online, Data Restore requests an exclusive lock on the dbspace. This lock ensures that no one can update data while Data Restore is processing the SELECT request. The exclusive dbspace lock may cause lock contention for other users. The SELECT function requires that you specify the SQLDBA password on the DBAPW parameter of the CONTROL statement as a security precaution.

When you specify OUTPUT=PRINTER, the SELECT function displays the selected data in a report format to SYSPRINT.

When the OUTPUT=TAPE or the OUTPUT=DASD parameter is specified, you can use the SELECT function to replace the DBSU DATAUNLOAD command. Data Restore 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. Examples are provided later in this chapter.

Note: The SELECT function divides at least by 2 the necessary time to unload data using the DBSU DATAUNLOD function.

Description

Figure 128 on page 148 shows an overview of the SELECT function. Make sure that the CREATOR and the TNAME parameters define a table, not a view. You can use the SELECT function to extract data from any table without LONG columns. Make sure you can access the **dbname** SQLDEF file on the database production disk.

Note: When you select the SELECT function, Data Restore does not execute local date and time user exits nor does it execute field procedures on the selected columns.

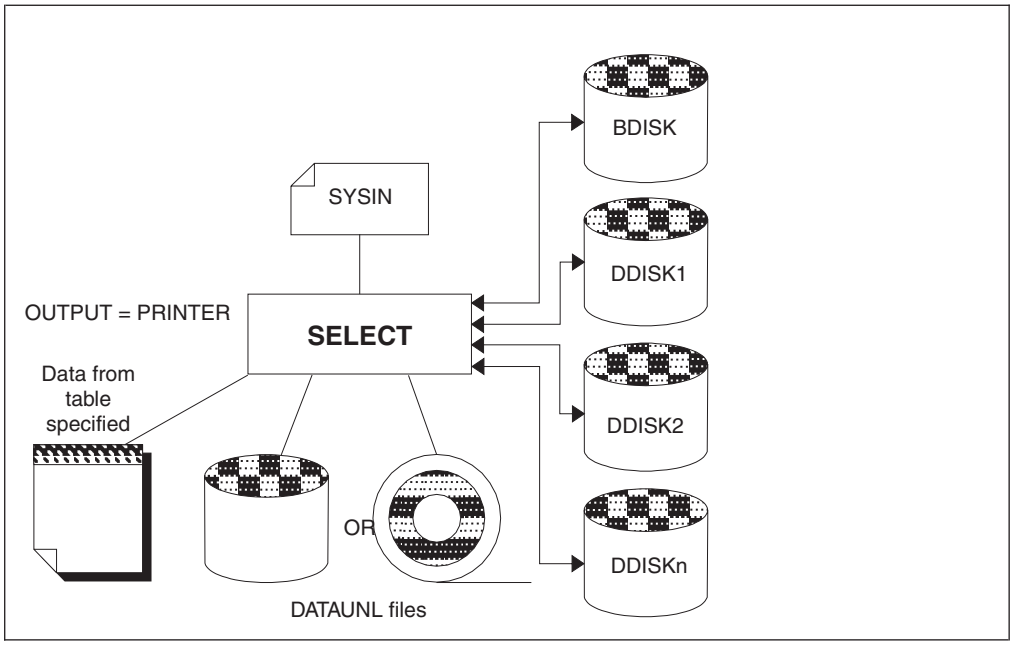


Figure 128. Overview of the SELECT function

SYSIN
Contains the name of the table to select, the names of the columns to select and the output device. You can use one SELECT statement in the SYSIN file.

SELECT
Is the function to process to access data.

OUTPUT
Contains the result of the SELECT function. The DBAPW parameter of the CONTROL statement appears as asterisks in the output.

DATAUNL files
Contain data when OUTPUT=TAPE/DASD is specified.

Using the SELECT Function in a VSE Environment

Figure 129 shows an example of JCL to execute the SELECT function.

```
        // JOB SELECT
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // EXEC PROC=ARIS73DB
(3) ---> // EXEC XTS91001,SIZE=AUTO
(4) ---> CONTROL DBAPW=XXXXXXXX
(5) ---> SELECT CREATOR=SQLDBA TNAME=CUSTOMERS
(6) --->  COLUMNS=(CUST_NO, CUST_NAME, CUST_ADDRESS)
(7) ---> /*
```

Figure 129. JCL to Execute a SELECT Function (VSE)

Statement 1 Specifies the DB2 library. You must specify the DB2 library for the SELECT function.

Statement 2 Uses an EXEC to access the database files.

Statement 3 Processes the function.

Statement 4 Specifies the SQLDBA password.

Statement 5 Indicates the creator name as well as the table name.

Statement 6 Indicates the column name criteria for selection.

Statement 7 Ends the SYSIN file.

Using the SELECT Function in a VM Environment

Figure 130 shows an EXEC to execute the SELECT function.

```
        /**/
(1) ---> 'FILEDEF SYSPRINT DISK SELECT SYSPRINT A'
(2) ---> 'FILEDEF SYSIN DISK SELECT SYSIN A'
(3) ---> 'XEDIT SELECT SYSIN A'
(4) ---> 'XTS91001'
```

Figure 130. Procedure to Execute a SELECT Function (VM)

Statement 1 Designates the SYSPRINT file to contain information about the process.

Statement 2 Specifies the SYSIN file.

Statement 3 Initiates an edit of the SYSIN file to allow for any changes before processing.

Statement 4 Starts the program to process the function.

The SYSIN file must contain the following statements:

```
(5) ---> CONTROL DBAPW=XXXXXXXX, DBNAME=dbname
(6) ---> SELECT CREATOR=SQLDBA TNAME=CUSTOMERS
(7) --->  COLUMNS=(CUST_NO, CUST_NAME, CUST_ADDRESS)
```

Figure 131. SYSIN File to Specify Table Selection

Statement 5 Specifies the database to process and the SQLDBA password.

Statement 6 Indicates the creator name and the table name.

Statement 7 Indicates the column name criteria for selection.

Make sure you can access the **dbname** SQLFDEF file on the database production disk when you execute Data Restore.

Displaying the Results of the SELECT Function

If you specify the SELECT function with the OUTPUT=PRINTER parameter, the number of rows selected is displayed on the console as shown in Figure 132 on page 150.

```
XTS9-143 CONTROL  DBAPW=********   DBNAME=dbname
XTS9-143 SELECT TNAME=CUSTOMER CREATOR=SQLDBA
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-309 Processing DB2 Server for VSE & VM version 7
XTS9-152      3 ROWS SELECTED
XTS9-007  PROCESSING SUCCESSFULLY COMPLETED
```

Figure 132. Console Messages for SELECT with the OUTPUT=PRINTER Parameter

Data Restore displays the result on SYSPRINT/SYSLST. Question marks (?) represent NULL columns as shown in Figure 133.

```
        TNAME=CUSTOMERS

COLNO  001   CNAME  "CUST_NO"
COLNO  002   CNAME  "CUST_NAME"
COLNO  003   CNAME  "CUST_ADDRESS"
COLNO  004   CNAME  "CUST_POSTCODE"
COLNO  005   CNAME  "CUST_TEL"
*******************************************************************************
1   *       2          * 3                                  *  4  * 5
*******************************************************************************
105725*DUPONT  PIERRE    *33 AV DES CHAMPS ELYSEES           *75008*44-40-41-42
309001*MARTIN  PAUL      *1 RUE DE LA PAIX                   *93001*37-37-10-12
25018 *DURAND JACQUES    *????????????????????????????????*75009*40-32-10-24
*******************************************************************************
```

Figure 133. Report for SELECT with OUTPUT=PRINTER Parameter

After a SELECT function with the OUTPUT=TAPE or the OUTPUT=DASD parameter, the number of rows selected is displayed on the console as shown in Figure 134.

```
XTS9-143 OPTIONS OUTPUT=DASD
XTS9-143 CONTROL  DBAPW=********   DBNAME=dbname
XTS9-143 SELECT TNAME=CUSTOMER CREATOR=SQLDBA
XTS9-143 /*
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-309 Processing DB2 Server for VSE & VM version 7
XTS9-152              3 ROWS SELECTED
XTS9-007    PROCESSING SUCCESSFULLY COMPLETED
```

Figure 134. Console Messages for SELECT with OUTPUT=DASD Parameter

Data Restore writes the data to tape or DASD and displays the column's position on the PRINTER as shown in Figure 135.

```
"CUST_NO"        007-010  FIXED
"CUST_NAME"      013-032  CHARACTER
"CUST_ADDRESS"   035-066  NULL IF POS(33)=255 CHARACTER
"CUST_POSTCODE"  069-073  NULL IF POS(67)=255 CHARACTER
"CUST_TEL"       076-085  NULL IF POS(74)=255 CHARACTER
```

Figure 135. List Displayed for SELECT with OUTPUT=DASD Parameter

Chapter 13. Displaying the Contents of an Archive File

Introduction

When you decide to restore a database or reload a table from an archive other than the current one, you must specify the correct version of the information you want to restore. Such information includes the date, time, and valid table names.

To obtain this information, use the DESCRIBE function which reads an archive or an unload file and produces a report with information about the contents of the file. From this report, you can determine the correct values for the required parameters.

Description

Figure 136 shows an overview of the DESCRIBE process. When reloading tables from an archive other than the latest, the *date* and *time* parameters on the CONTROL statement are required. When reloading from the latest archive, these parameters are optional. Use the DESCRIBE function to list all the information about the archive file: date, time, and names of reloadable tables. Figure 137 on page 152 shows an example of using the DESCRIBE function. Use the DESCRIBE function to list all reloadable tables from an archive or an UNLOADed dbspace.

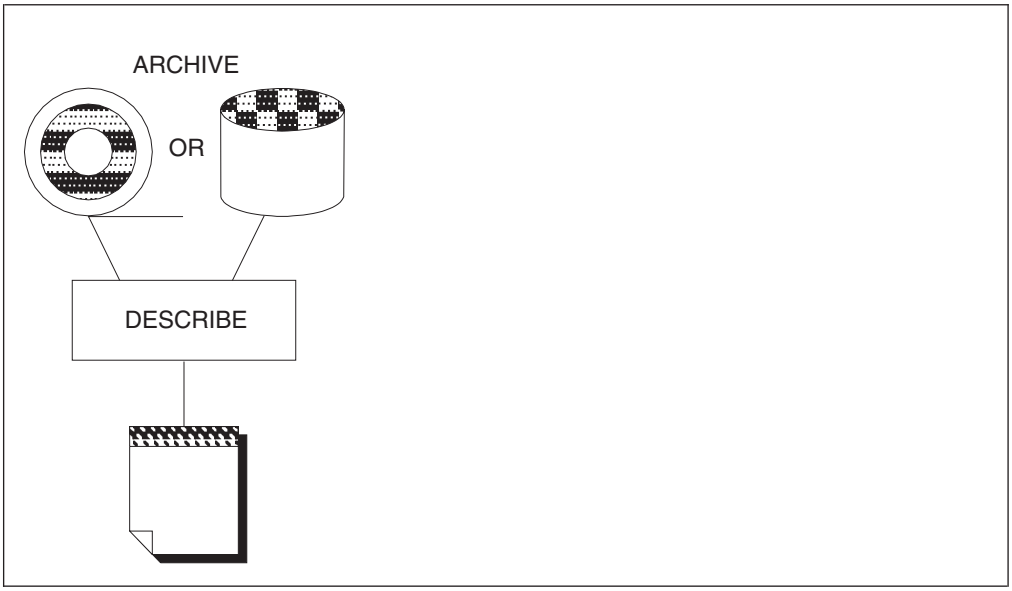


Figure 136. Overview of the DESCRIBE Process

Using the DESCRIBE Command

Figure 138 on page 152 shows how to use the DESCRIBE function in a VM environment.

```
        // JOB DESCRIBE
(1) ---> // LIBDEF *,SEARCH=(PRD2.RCVvrm)
(2) ---> // TLBL ARCHIV,'ARCHIVE.DB2'
(3) ---> // ASSGN SYS006,180
(4) ---> // MTC REW,SYS006
(5) ---> // EXEC XTS91001,SIZE=AUTO
(6) ---> DESCRIBE
(7) ---> /*
```

Figure 137. JCL for the DESCRIBE function in VSE

Statement 1 Specifies the Data Restore library. You must specify the Data Restore library for the DESCRIBE function.

Statement 2 Identifies the label on the tape.

Statement 3 Assigns the tape drive.

Statement 4 Rewinds the tape to the first file.

Statement 5 Executes the program XTS91001.

Statement 6 Performs the DESCRIBE function as requested.

Statement 7 Ends the SYSIN file.

Sample Procedure (VM).

```
        /**/
(1) ---> 'TAPE REW'
(2) ---> 'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
(3) ---> 'FILEDEF SYSPRINT DISK XTS91001 SYSPRINT A'
(4) ---> 'FILEDEF SYSIN DISK RESTORE SYSIN A'
(5) ---> 'XTS91001'
```

Figure 138. EXEC for the DESCRIBE Function in VM

Statement 1 Rewinds the tape to its first file.

Statement 2 Identifies the label on the tape.

Statement 3 Specifies the destination of the SYSPRINT file.

Statement 4 Identifies the input source for the DESCRIBE function.

Statement 5 Executes the program XTS91001.

The SYSIN file must contain the following statements:

```
(6) ---> DESCRIBE
```

Figure 139. Input Statements for the DESCRIBE Function

Statement 6 Specifies the function to perform. In this case, the DESCRIBE function.

Figure 140 on page 153 shows an example of a report after completing the DESCRIBE function on a DB2 FULL archive. Message XTS9-136 shows that the archive in the example was taken on February 10th, 1997 at 09:03:11. Message

XTS9-007 at the end of the report indicates that the DESCRIBE function executed successfully. The rest of the report lists the tables that you can reload.

```
    XTS9-143 CONTROL BASE=SQLDBA
    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 VERSION 7.1.0
    XTS9-136 Processing SQLDBA  archived on (02/10/97-09:03:11)
--> XTS9-229 The file is a DB2 FULL archive
    XTS9-013 Table SQLDBA  .ACTIVITY     may be reloaded
    XTS9-013 Table SQLDBA  .DEPARTMENT   may be reloaded
    XTS9-007 Processing successfully completed
```

Figure 140. Output from the DESCRIBE Function Using a DB2 FULL Archive

Figure 141 shows an example of a report after completing the DESCRIBE function on a DB2 regular archive.

```
    XTS9-143 CONTROL BASE=SQLDBA
    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 VERSION 7.1.0
    XTS9-136 Processing SQLDBA  archived on (02/10/97-09:43:25)
--> XTS9-229 The file is a DB2 regular archive
    XTS9-013 Table SQLDBA  .ACTIVITY     may be reloaded
    XTS9-013 Table SQLDBA  .DEPARTMENT   may be reloaded
    XTS9-007 Processing successfully completed
```

Figure 141. Output from the DESCRIBE Function Using a DB2 FULL Archive

You can use the DESCRIBE function to get information on the date and time of a previous archive. Figure 142 and Figure 143 on page 154 show how to use the DESCRIBE function in the VSE and VM environments.

```
        // JOB DESCRIBE
(1) ---> // LIBDEF *,SEARCH=(PRD2.RCVvrm)
(2) ---> // TLBL ARCHIV,'ARCHIVE.DB2'
(3) ---> // ASSGN SYS006,180
(4) ---> // MTC REW,SYS006
(5) ---> // EXEC XTS91001,SIZE=AUTO
        CONTROL
(6) ---> DESCRIBE
(7) ---> /*
```

Figure 142. JCL to View the Contents of a Tape (VSE)

Statement 1 Specifies the DB2 for VSE Library for the DESCRIBE function. (You must specify the DB2 library.)

Statement 2 Identifies the label on the tape.

Statement 3 Assigns the tape drive.

Statement 4 Rewinds the tape to the first file.

Statement 5 Executes the program XTS91001.

Statement 6 Processes the DESCRIBE function as requested.

Statement 7 Ends the SYSIN.

```
        /**/
(1) ---> 'TAPE REW'
(2) ---> 'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
(3) ---> 'FILEDEF SYSPRINT DISK XTS91001 SYSPRINT A'
(4) ---> 'FILEDEF SYSIN DISK DESCRIBE SYSIN A'
(5) ---> 'XTS91001'
```

Figure 143. EXEC to List the Contents of a Tape (VM)

Statement 1 Rewinds the tape to its first file.

Statement 2 Identifies the label on the tape.

Statement 3 Specifies the destination of the SYSPRINT file.

Statement 4 Identifies the input source for the DESCRIBE function.

Statement 5 Executes the program XTS91001.

The SYSIN file must contain the following statements:

```
(6) ---> CONTROL DBNAME=dbname
(7) ---> DESCRIBE
```

Figure 144. SYSIN File to List the Contents of a Tape

Statement 6 Points to the appropriate database.

Statement 7 Specifies the function to be performed. In this case, the DESCRIBE function.

Use the job (or a similar one) as depicted in the preceding figures.

Chapter 14. Displaying Dbspace Information

Introduction

It can be most useful to you, the database administrator, if you can obtain current information relating to the dbspace. The SHOWDBS function provides information such as the number of header, data, or index pages available in a dbspace when the server is offline or online.

The SHOWDBS command is similar to the DB2 SHOW DBSPACE operator command. The SHOWDBS function usually provides better performance than the SHOW DBSPACE command.

Note: Only information about acquired dbspaces is displayed.

Description

Figure 145 depicts the SHOWDBS process.

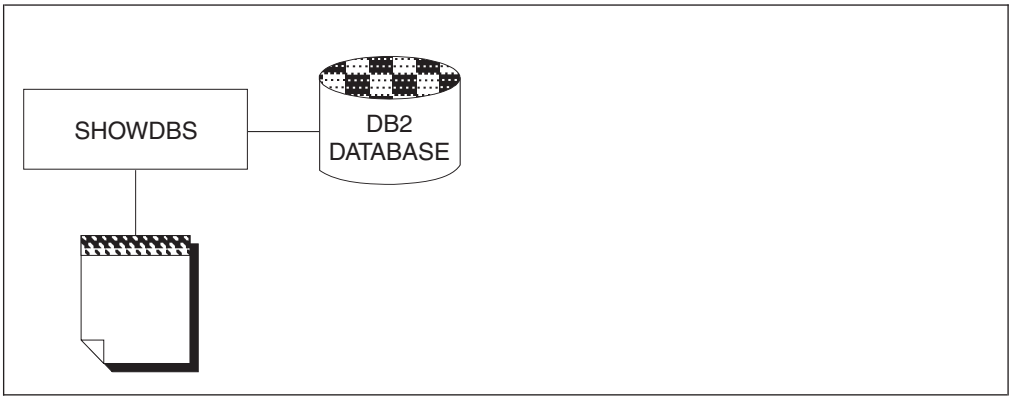


Figure 145. Overview of the SHOWDBS Function

Using the SHOWDBS Command

The following is an example of the JCL required in VSE:

```
         // JOB SHOWDBS
(1) ---> // LIBDEF *,SEARCH=(PRD2.RCVvrm)
(2) ---> // EXEC PROC=ARIS73DB
(3) ---> // EXEC XTS91001,SIZE=AUTO
(4) ---> // MTC REW,SYS006
(5) ---> OPTIONS NOTATION=U
(6) ---> SHOWDBS SORT=NAME
(7) ---> /*
```

Figure 146. JCL to Execute the SHOWDBS command

Statement 1 Specifies the Data Restore library. You must specify the Data Restore library for the SHOWDBS function.

Statement 2 Contains all DLBL for the dbextents.

Statement 3	Executes the program XTS91001.
Statement 4	Rewinds the tape to the first file.
Statement 5	Allows the user to indicate the notation preference, United States or European.
Statement 6	Performs the SHOWDBS function as requested while sorting the output by dbspace name.
Statement 7	Ends the SYSIN file.

The following is an example of an EXEC to use in VM:

```
        /**/
(1) ---> 'FILEDEF SYSPRINT DISK XTS91001 SYSPRINT A'
(2) ---> 'FILEDEF SYSIN DISK SHOWDBS SYSIN A'
(3) ---> 'XTS91001'
```

Figure 147. VM EXEC to Run the SHOWDBS function

Statement 1	Specifies the destination of the SYSPRINT file.
Statement 2	Identifies the input source for the SHOWDBS function.
Statement 3	Executes the program XTS91001.

The SYSIN file must contain the following statements:

```
(4) ---> OPTIONS NOTATION=U
(5) ---> CONTROL DBNAME=dbname
(6) ---> SHOWDBS SORT=NAME
```

Figure 148. Input File to Run the SHOWDBS function

Statement 4	Allows you to indicate your notation preference, United States or European.
Statement 5	Specifies the database name.
Statement 6	Performs the SHOWDBS function as requested while sorting the output by dbspace name.

Report for SHOWDBS Function

```
XTS9-143 OPTIONS NOTATION=U
XTS9-143 CONTROL   DBNAME=dbname
XTS9-143 SHOWDBS   SORT=NAME
XTS9-143 /*
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-313 DB2 for VSE & VM version 7 processed.
XTS9-601 DATE:11/10/94  TIME:16:18:05
--------------------------------------------------------------------------------
XTS9-014 Pool   1 Dbno   4 Name TEST     Pages     1,024 Headr 8 Index  33

XTS9-019                                Header        Data        Index
XTS9-020                               ---------   --------   ---------
XTS9-018 Maximum                           8          679         337
XTS9-016 Used                              1            4           3
XTS9-017 %Used                          12.5          0.5         0.8
--------------------------------------------------------------------------------
XTS9-014 Pool   1 Dbno   8 Name TEST2    Pages     1,024 Headr 8 Index   33

XTS9-019                                Header        Data         Index
XTS9-020                               ----------   ---------     ---------
XTS9-018 Maximum                           8          679         337
XTS9-016 Used                              1            5          15
XTS9-017 % Used                         12.5          0.7         4.4
--------------------------------------------------------------------------------
XTS9-014 Pool  -2 Dbno  27 Name XTEST1   Pages       256 Headr 8 Index  33

XTS9-019                                Header        Data        Index
XTS9-020                               ---------     --------   ---------
XTS9-018 Maximum                           8          164          84
XTS9-015 Reserved                          1            1           1
XTS9-016 Used                              1            0           0
XTS9-017 %Used                          12.5          0.0         0.0
--------------------------------------------------------------------------------
```

Figure 149. Report for the SHOWDBS function

Report for SHOWDBS Function (Summary Report)

```
XTS9-601 DATE:11/10/94  TIME:16:18:05
XTS9-022 Pool:001
XTS9-023 ========
XTS9-024                Dbno   Dbspacename        Reserved       Used      Empty
XTS9-025               -----   ------------      ---------     ------     -------
XTS9-021                1       XTEST2            2,622         171       2,451
XTS9-021                2       ZTEST               404         318          86
XTS9-026                                          --------     ------     -------
XTS9-027 Total pool:  1                            3,026         487       2,537
XTS9-022 Pool:002
XTS9-023 ========
XTS9-024                Dbno     Dbspacename      Reserved       Used      Empty
XTS9-025               -----    -------------     ---------      -----    -------
XTS9-021                 27      XTEST1                3           1          2
XTS9-026                                          ---------      ------   -------
XTS9-027 Total pool: -2                                3           1          2
XTS9-601 DATE:11/10/94  TIME:16:18:05
XTS9-007 Processing successfully completed
```

Figure 150. Summary Report for the SHOWDBS Function

Interpreting the SHOWDBS Reports

The SHOWDBS function produces two reports: a detailed report for all acquired dbspaces, and a summary report for dbspaces with empty pages.

The detailed report displays:

1. Information about the storage pool:

   XTS9-014 Pool **N1** Dbno **N2** Name **N3** Pages **N4** Heardr **N5** Index **N6**

   ```
   N1 : Pool number (if negative, this is nonrecoverable)
   N2 : Dbspace number
   N3 : Dbspace name
   N4 : Total number of pages in the dbspace (NPAGES)
   N5 : Number of header pages (NHEADER)
   N6 : Percentage of pages reserved for index pages in the dbspace (PCTINDEX)
   ```

2. Information about maximums for the dbspace:

   ```
   XTS9-018 Maximum   N1        N2        N3
   N1 : Maximum number of header pages
   N2 : Maximum number of data pages
   N3 : Maximum number of  index pages
   ```

3. Information about reserved pages in the pool for the dbspace:

   Note: These reserved pages cannot be used by any other dbspace in the storage pool.

   ```
   XTS9-015 Reserved  N1        N2        N3
   N1 : Number of reserved header pages
   N2 : Number of reserved data pages
   N3 : Number of reserved index pages
   ```

   This information is displayed only if there are reserved pages.

4. Information about used pages in the dbspace:

   ```
   XTS9-016 Used  N1       N2        N3
   N1 : Number of used header pages
   N2 : Number of used data pages
   N3 : Number of used index pages
   ```

5. Information about percentage of used pages:

   ```
   XTS9-017 % Used  N1        N2        N3
   N1 : Value of percentage of used header pages
   N2 : Value of percentage of used data pages
   N3 : Value of percentage of used index pages
   ```

   Note: Percentage is obtained by:

% USED = (USED PAGES/MAXIMUM PAGES) * 100

The summary report displays all dbspaces with empty pages. The dbspaces are grouped by storage pool.

1. Information about the storage pool:

   ```
   XTS9-022 Pool: N1
   N1 : Storage pool number
   ```

2. Information about the dbspaces with empty pages:

   ```
   XTS9-024  Dbno  Dbspacename  Reserved   Used   Empty
   XTS9-021  N1    N2          N3         N4     N5
   ```

```
N1 : Dbspace number
N2 : Dbspace name
N3 : Number of reserved pages
N4 : Number of used pages
N5 : Number of empty pages (N3-N4)
```

3. Information about storage pools having at least one dbspace with empty pages:

```
XTS9-027  Total pool: N1        N2          N3        N4
N1 : Pool number (if negative, the storage pool is non-recoverable)
N2 : Total number of reserved pages in the pool for dbspaces with empty pages
N3 : Total number of used pages in the pool for dbspaces with empty pages
N4 : Total number of empty pages in the pool for dbspaces with empty pages (N2-N3)
```

Chapter 15. Displaying Pool Organization

Introduction

The physical organization of a database is composed with dbextents, VSAM clusters for DB2 Server for VSE or minidisks for DB2 Server for VM those dbextents are assigned to a storage pool(STORPOOL) during database generation or during ADD DBEXTENT function.

If a disk on which the database is defined is damaged, it is necessary to restore the lost information. You can restore the whole database or decide to restore only the affected pool. When performing storage pool level recovery, the SHOWPOOL function is used to identify which pools must be restored by listing the dbextents associated with each storage pool defined on the database server.

Description

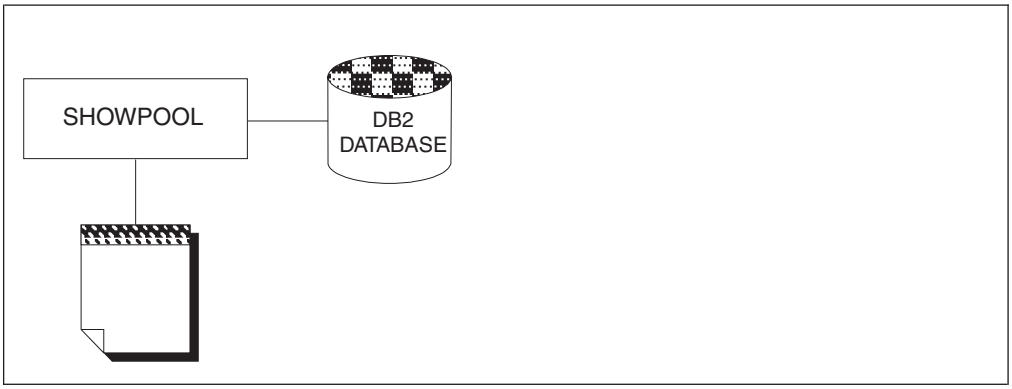


Figure 151. General process for SHOWPOOL Function

Using the SHOWPOOL Command

The following is an example of using JCL to execute the SHOWPOOL command.

```
        // JOB SHOWPOOL
(1) ---> // LIBDEF *,SEARCH=(PRD2.RCVvrm)
(2) ---> // EXEC PROC=ARIS73DB
(3) ---> // EXEC XTS91001,SIZE=AUTO
(4) ---> SHOWPOOL
        /*
```

Figure 152. Sample of JCL (VSE) to execute SHOWPOOL

Statement 1	Specifies the Data Restore library. You must specify the Data Restore library for the SHOWPOOL function.
Statement 2	Contains all DLBL for DBEXTENTS.
Statement 3	Executes the program XTS91001.
Statement 4	Process the SHOWPOOL command.

Sample Procedure(VM)

```
        /**/
(1) ---> 'FILEDEF SYSPRINT DISK XTS91001 SYSPRINT A'
(2) ---> 'FILEDEF SYSIN DISK SHOWPOOL SYSIN A'
(3) ---> 'XTS91001'
```

Figure 153. Sample VM procedure for running the SHOWPOOL function

Statement 1 Specifies the destination of the SYSPRINT file.

Statement 2 Identifies the input source for the SHOWPOOL function.

Statement 3 Executes the program XTS91001.

The SYSIN file must contain the following statements:

```
(4) ---> CONTROL DBNAME=dbname
(5) ---> SHOWPOOL
```

Figure 154. Sample input file for running the SHOWPOOL function

Statement 4 Specifies the database name.

Statement 5 performs the SHOWPOOL function.

Report for SHOWPOOL Function

```
XTS9-143 CONTROL BASE=BASE2
XTS9-143 SHOWPOOL
XTS9-143 /*
XTS9-196 Do you want to continue the SHOWPOOL process ?
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-313 DB2 for VSE & VM version 7 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=    4      pool is RECOVERABLE
XTS9-143 ************************************************************
XTS9-204 Dbextent=     1  Pool=  1  Next dbextent=NONE
XTS9-204 Dbextent=     2  Pool=  2  Next dbextent=      3
XTS9-204 Dbextent=     3  Pool=  2  Next dbextent=NONE
XTS9-204 Dbextent=     4  Pool=  3  Next dbextent=NONE
XTS9-204 Dbextent=     5  Extended deleted
XTS9-143 ************************************************************
XTS9-007 Processing successfully completed
```

Figure 155. Sample of Report for SHOWPOOL Function

Interpreting the Output From the SHOWPOOL Function

The SHOWPOOL command produces a report consisting of two parts.

The first part displays, for each pool, the first dbextent of the pool, and if the pool is recoverable or not.

The second part displays each dbextent, the pool in which it is located, and the next dbextent in the same pool.

The information in the first part of the report has the following format:

XTS9-203 Pool=**N1** first dbextent=**N2** pool is **N3**

N1 is the pool number
N2 is the number of the first dbextent in pool N1
N3 indicates if the pool is RECOVERABLE or NON-RECOVERABLE

The information in the second part of the report has the following format:

XTS9-204 dbextent=**N1** pool=**N2** next dbextent=**N3**

N1 is the dbextent number
N2 is the pool number for dbextent N1
N3 is the next dbextent number in the pool or
NONE, indicating that this is the last dbextent in the pool

Using the SHOWPOOL report

The SHOWPOOL report is used to verify which storage pool should be recovered in case of a DASD failure.

The second part of the report is used to determine which storage pool contains the affected dbextent(s). The first part of the report is then used to determine if that storage pool is eligible for storage pool level recovery (indicated by 'pool is RECOVERABLE' in message XTS9-203.) The pool number can then be included in the pool list for the RESTORE function.

Part 3. Reference

This part contains the following information:

- Command syntax
- Options and control statements
- Performance and tuning
- Messages and codes
- Examples of commands
- Problem determination

Chapter 16. Command Syntax

All commands start in column 1. If the statement is longer than one line, continue the statement on the next line starting after a column position other than 1.

You can use a mixture of both upper or lower case.

To include comments in the SYSIN file, enter an asterisk (*) as the first character on the line.

You can separate parameters with either a space or comma.

The SYSIN report is sent to the printer device and shows all the input control records used to execute the function.

An example SYSIN file that processes a Data Restore function is shown in Figure 156. (This example reloads a table named CUSTOMERS created by SQLDBA.)

```
(1) ---> OPTIONS DEVICE=TAPE LANG=S002
(2) ---> CONTROL DBNAME=SQLDBA
(3) ---> RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS
(4) --->   FUNCT=NEW
(5) ---> * RELOAD CREATOR=SQLDBA TNAME=EMPLOYEE
```

Figure 156. Example SYSIN File to Process a Data Restore Function

Statement 1 Specifies the OPTIONS statement starts in column 1. You must define the OPTIONS statement on the first line of the SYSIN file.

Statement 2 Specifies the CONTROL statement which starts in column 1 and must follow the OPTIONS statement.

Statement 3 Identifies the function starting in column 1. The function definition must follow the OPTIONS and CONTROL statements.

Statement 4 Continues the statement started on the preceding line. Note that the statement does not start in column 1.

Statement 5 Uses an asterisk in column 1 to indicate a comment.

OPTIONS and CONTROL Statements

You can specify additional parameters for each of the Data Restore functions. Specify these parameters in the OPTIONS statement or the CONTROL statement.

The list of those parameters follows:

OPTIONS Statement Parameters

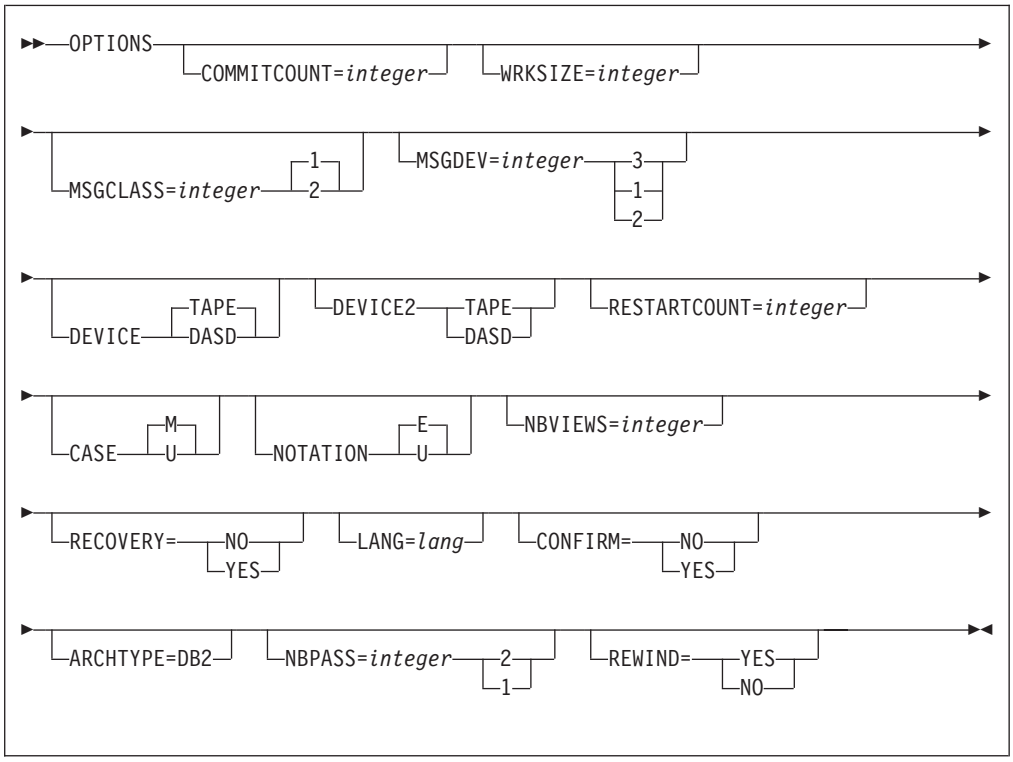


Figure 157. OPTIONS statement

COMMITCOUNT
When the RELOAD or APPLYLOG function is invoked, this parameter executes a COMMIT WORK after the specified number of rows has been reloaded (no default value).

WRKSIZE Available memory for the Data Restore feature to work, expressed in KB (default value is 2048KB for BACKUP function and 256KB for any other function). Take WKSIZE= NBP*5/1024 where NBP is the number of active pages of the database to be unloaded or backed up. If WKSIZE is not great enough Data Restore, at execution time, will send message XTS9-122 followed by XTS9-158 containing the value needed.

When reloading or restoring from an incremental backup, the WRKSIZE parameter must be equivalent to the value specified at backup time. Message XTS9-230 will display the required value.

MSGCLASS **1:** All messages are displayed. This is the default.

2: Only error messages are displayed

MSGDEV **1:** Messages are displayed on VSE SYSLOG or VM console

2: Messages are displayed on VSE SYSLST or VM SYSPRINT

3: Messages are displayed on both devices

DEVICE **TAPE:** Archive on tape. This is the default.

DASD: Archive on disk

The following parameter, DEVICE2, is optional for the BACKUP or the UNLOAD function should the user want to perform a primary and a secondary backup of the archive in a single operation:

DEVICE2 For BACKUP Function

TAPE: A second archive copy is made on tape

DASD: A second archive copy is made on disk (ARCHIV2 must be specified in JCL or EXEC).

In the VM environment, this parameter only specifies that dual archiving is enabled. (It does not matter if the parameter is set to tape or DASD.) The FILEDEF for ARCHIV2 will determine whether tape or DASD output is created.

In the VSE environment, if tape is specified a TLBL ARCHIV2 must be defined. If DASD is specified, a DLBL must be specified for ARCHIV2.

DEVICE2 For RELOAD or RESTORE From an Incremental Backup File

TAPE: FULLARC TLBL must be specified for FULL backup file

DASD: FULLARC DLBL must be specified for FULL backup file

RESTARTCOUNT

When the RELOAD function is invoked, this parameter specifies how many rows will be skipped before reloading. This parameter is used for recovery after reload error, for that reason all reload cards in SYSIN **must not be modified**.

CASE

M: Mixed to display messages with lower and uppercase. This is the default.

U: Upper to display messages with upper case only, (for printers not supporting lower case).

LANG To obtain Data Restore feature messages in a specific language, specify the National Language code on LANG parameter:

S001 - American English (AMENG). This is the default.
S002 - Upper case English
S003 - French
S004 - German
D001 - Japanese
D003 - Chinese

NOTATION

E: European: Numbers are displayed with point for thousand and comma for decimal. For example: 1.234.567,89. This is the default.

U: United States: Numbers are displayed with point for decimal and comma for thousand. For example: 1,234,567.89

RECOVERY

NO: No table recovery from log files is required. This is the default.

YES: The user wants to generate the file containing all modifications on reloaded tables from LOGS.

NBVIEWS If the database contains many views (more than 200 for the

reloaded tables) the user will specify the number of views estimated to recreate after RELOAD function (default value: 200).

CONFIRM Only available in VM environment

YES: Before processing the function in SYSIN, Data Restore will prompt the operator to confirm the function to

NO: No confirmation for the process is required.

ARCHTYPE **DB2:** The RELOAD function is to be processed directly from a standard DB2 Server for VSE & VM archive file.

NBPASS For RELOAD processing from DB2 Server for VSE & VM archive indicates the number of reading passes.

2: For normal processing. This is the default.

1: can be specified on a Restart (for instance after a power failure) to avoid reading the DB2 Server for VSE & VM archive twice, only valid if the message XTS9-193: ″Mount first tape of DB2 Server for VSE & VM archive″ was previously received.

REWIND For UNLOAD processing in VSE only, whether the output tape should be rewound during OPEN/CLOSE processing.

YES: Rewind during OPEN/CLOSE processing.

NO: Do not rewind during OPEN/CLOSE processing.

CONTROL Statement Parameters

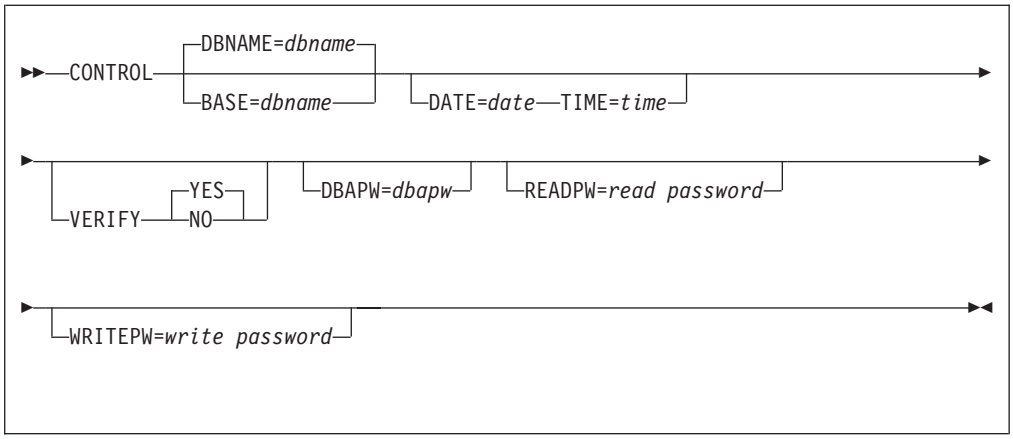


Figure 158. CONTROL statement

DBNAME Dbname of database to process. This parameter is optional for the VSE environment (BASE is equivalent to DBNAME parameter).

DATE Date of BACKUP.

This parameter must be specified when restoring from an archive that is not the last archive executed. The date format is provided by your system.

TIME Time of BACKUP.

This parameter must be specified when restoring from an archive that is not the last archive executed. The time format is HH:MM:SS.

VERIFY With VERIFY=YES parameter: if the parameter DBNAME on the CONTROL statement is different from the archive TAPE/DASD the RELOAD job terminates immediately. (default value).

With VERIFY=NO parameter: even if the parameter DBNAME on the CONTROL statement is different from the archive TAPE/DASD, the process continues.

DBAPW RELOAD function needs DBA authority to execute all commands necessary to restore a table. A connect to SQLDBA is issued during processing. If SQLDBA user's password has changed between BACKUP and RELOAD function, the new password must be specified on CONTROL statement.

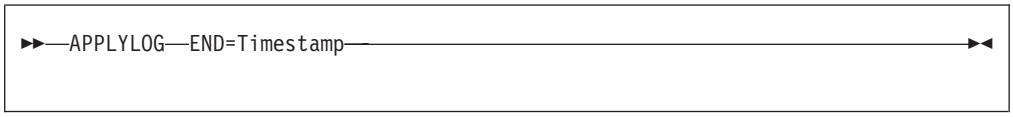
This parameter must be specified for SELECT and UNLOAD functions.

It will appear as asterisks on output for confidentiality reasons.

READPW Data Restore links application server's minidisks during processing, if the minidisks are password protected, the READ password must be specified (VM only) for SELECT, UNLOAD, RELOAD and SHOWDBS functions (all minidisks must be defined with the same password).

WRITEPW Data Restore links application server's minidisks during processing, if the minidisks are password protected, the WRITE password must be specified (VM only) for BACKUP and RESTORE functions (all minidisks must be defined with the same password).

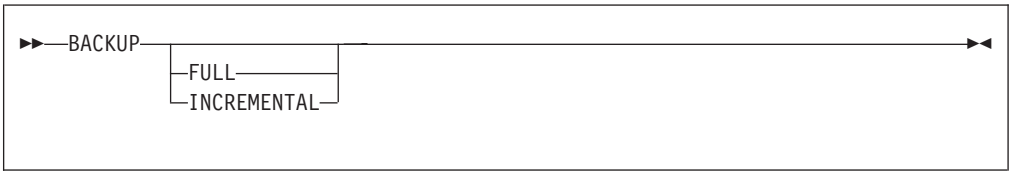
APPLYLOG



Purpose

Use the APPLYLOG command to apply the DB2 statements that were extracted from the log (and the log archive) during the RELOAD operation.

BACKUP



Purpose

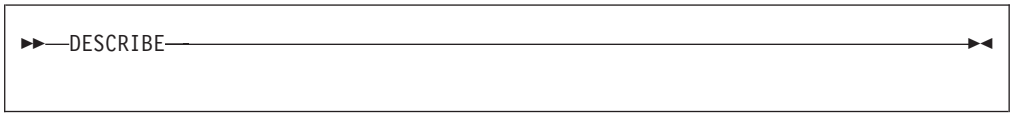
Use the BACKUP command to back up your database.

Operands

FULL
A full backup is executed using this parameter.

INCREMENTAL
An incremental backup is executed using this parameter. NOTE: This parameter can only be specified if a FULL backup has been executed.

DESCRIBE



Purpose

Use the DESCRIBE command to read an archive or unload a file and produce a report with information about the contents of the file.

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 archive identification must be specified in the CONTROL statement with "DATE=" and "TIME=" for the Data Restore RESTORE. Refer to Figure 168 on page 205 for a sample CONTROL statement that defines an archive to be restored. 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:

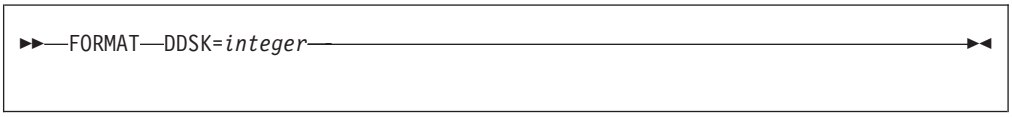
For **VM** only:

- **SYSIN** file - contains the command to be processed; in this case the DESCRIBE function. Figure 164 on page 204 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 166 on page 204 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

FORMAT



Purpose

Use the FORMAT command to format a VSE/VSAM defined file in the DB2 Server for VSE required 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:

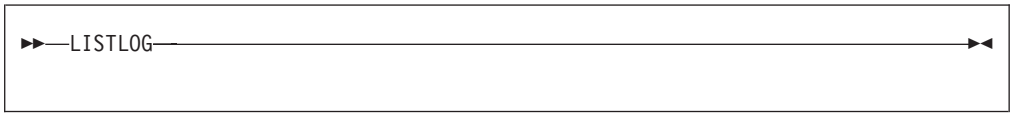
- the directory disk (BDISK). The Data Restore feature formats the BDISK while restoring the database.
- the log disk (LOGDSK1). To format the log disk, the COLDLOG procedure with the parameters SYSMODE=S and STARTUP=L should be used.

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.

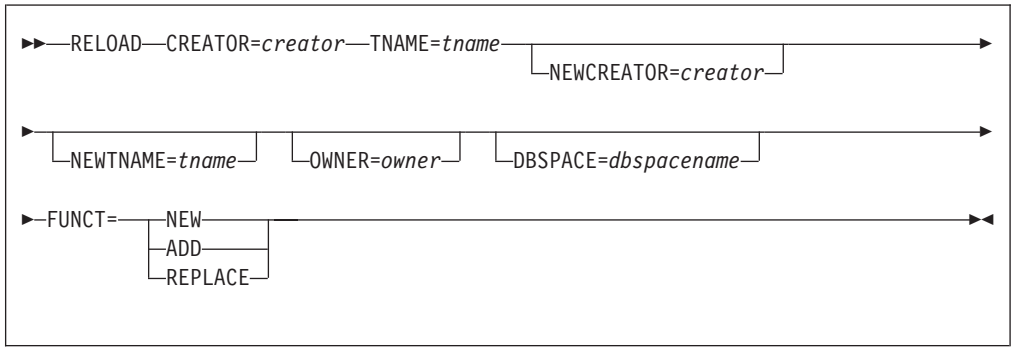
LISTLOG



Purpose

To list the contents of LMBRLG1, LMBRLG2, and LMBRLG3, execute the LISTLOG function using the XTS91001 program.

RELOAD



Purpose

Use the RELOAD command to reload tables.

Operands

RELOAD
Executes a reload of a table.

FUNCT
Specifies the reload option to use:

PURGE
Deletes all rows from the table NEWCREATOR.NEWTNAME and restores all the rows from the table CREATOR.TNAME into it.

NEW
Creates the table NEWCREATOR.NEWTNAME and restores all the rows from CREATOR.TNAME into it.

ADD
Inserts all the rows from CREATOR.TNAME into the table NEWCREATOR.NEWTNAME.

REPLACE
Drops and recreates the table NEWCREATOR.NEWTNAME, restores all the rows from the table CREATOR.TNAME into it, and recreates all the indexes, referential integrity, views, grants, comments and labels.

Note: This option creates the table even if the table did not exist previously.

TNAME
Specifies the name of the table to be restored.

CREATOR
Specifies the creator of the table to be restored.

NEWTNAME
Specifies a new table name, in case you want to reload a table with a different name.

NEWCREATOR
Specifies a new creator for the table, in case you want to reload a table with a different creator.

The CREATOR.TNAME table is reloaded into NEWCREATOR.NEWTNAME.

If you do not specify these optional parameters, Data Restore uses TNAME for NEWTNAME and CREATOR for NEWCREATOR, so the table will be reloaded with the same name.

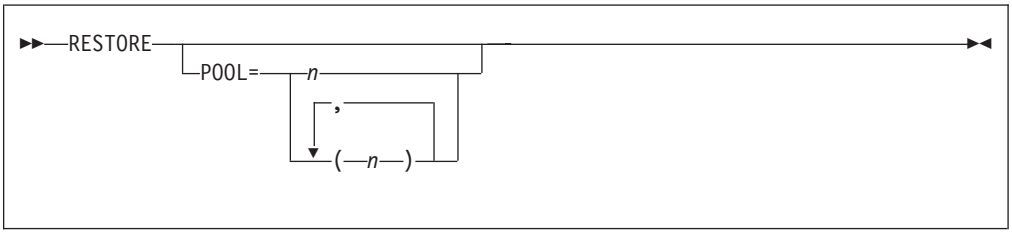
DBSPACE
Specifies the name of the dbspace which will contain the table to be restored. This parameter must be specified for FUNCT=NEW and FUNCT=REPLACE. If the table CREATOR.TNAME exists, the default value for the parameter is the name of the dbspace where this table resides.

OWNER
Specifies the owner of the dbspace which contains table to be restored. If you do not specify an owner, PUBLIC is assumed. This parameter must be specified for private dbspaces.

Usage Notes

1. If the table to be reloaded contains a LONG column, a file called LMBRWRK is created and used as a workfile. LMBRWRK contains all of the data pages of the dbspace containing the table with LONG columns.
2. If the first character on the SYSIN card is *, the card not used. The SYSIN is displayed on the PRINTER.
3. An OPTIONS statement can also be specified (for more information see "OPTIONS and CONTROL Statements" on page 167).
4. To reload more than one table, specify more RELOAD functions in SYSIN (up to 90 tables can be reloaded in one SYSIN).
5. To list the tables that can be reloaded from a BACKUP or UNLOAD file, use the DESCRIBE function (refer to Chapter 13, "Displaying the Contents of an Archive File" on page 151 for details).

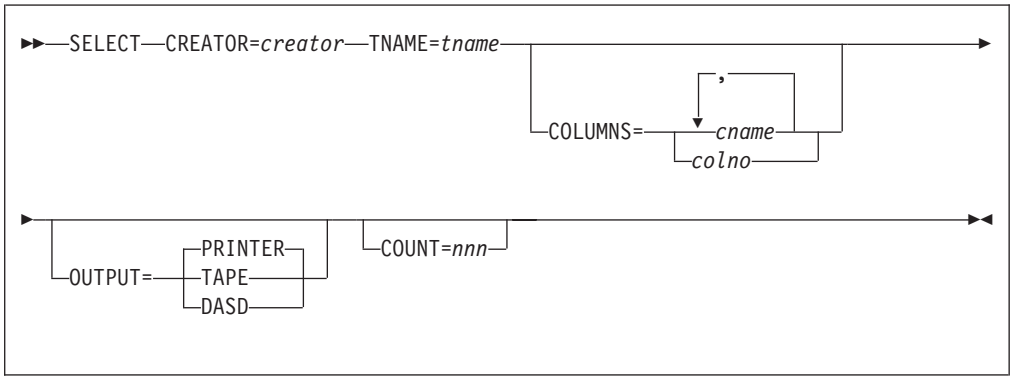
RESTORE



Purpose

Use the RESTORE command 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.

SELECT



Purpose

The SELECT command lets you select data from DB2 tables directly out of the dbextents, bypassing the database manager.

Operands

CREATOR
The name of the creator of the table.

TNAME
The name of the table which will be selected

COLUMNS
The specific columns to be selected. It can list either column name(s) or column number(s). If you omit the COLUMNS= parameter, all columns are selected.

OUTPUT
The output device:

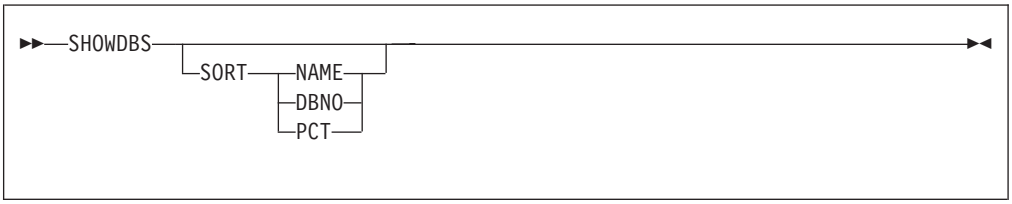
PRINTER The output file is spooled to the printer. This is the default.

TAPE The output file is a tape file

DASD The output file is a disk file

COUNT
The specific number of rows to be selected (from 1 to 32000).

SHOWDBS



Purpose

Use the SHOWDBS command to provide information such as the header, data or index pages available in a dbspace when the server is offline or online.

The Data Restore SHOWDBS executes a SHOW DBSPACE function while the database is online or offline.

Operands

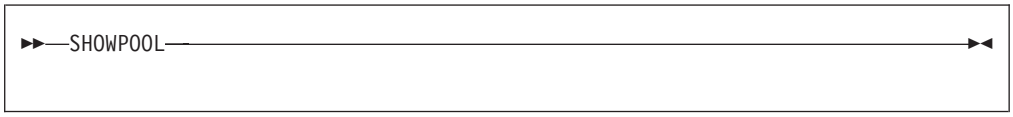
SORT
Use this parameter to obtain sorted output. This parameter is optional.

PCT
Displays the results sorted by dbspaces having the highest percentage of used data or index pages.

NAME
Displays results sorted by dbspace name.

DBNO
Displays the results sorted by dbspace number.

SHOWPOOL

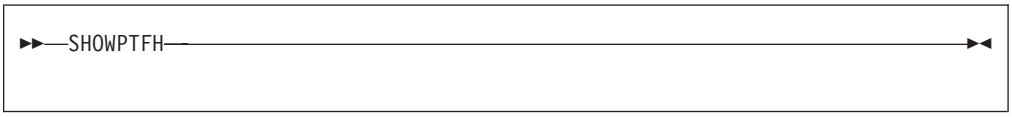


Purpose

Use the SHOWPOOL command to identify which pools must be restored by listing the dbextents associated with each storage pool defined on the database server.

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.

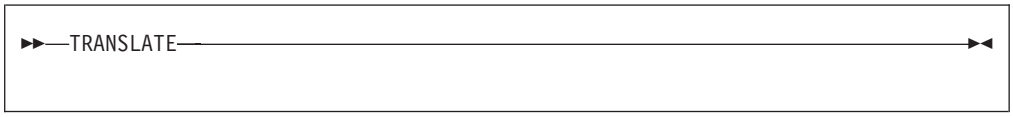
SHOWPTFH



Purpose

Use the SHOWPTFH command to display all Program Temporary Fixes (PTF) installed on the system for Data Restore feature.

TRANSLATE



Purpose

Use the TRANSLATE command to increase the performance when later restoring the complete database or parts of it from a DB2 archive.

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

Figure 159 on page 185 is an example of a function report for the TRANSLATE function on a DB2 FULL archive.

```
     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 7.1.0
-->  XTS9-229 The file is a DB2 FULL archive
     XTS9-193 Mount first tape of a 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 SQLDBA     .DEPARTMENT   MAY BE RELOADED
     XTS9-007 Processing successfully completed
```

Figure 159. Sample Report after Translation of a DB2 Server for VM FULL Archive

Figure 160 is an example of a function report for the TRANSLATE function on a DB2 regular archive.

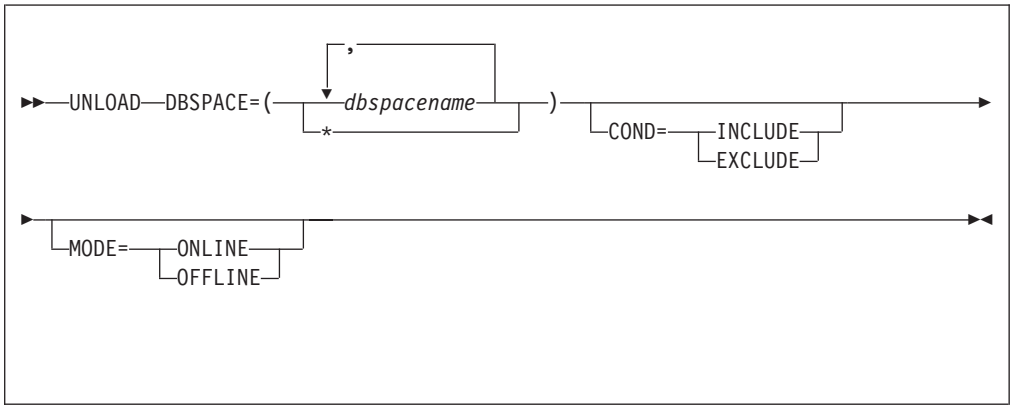
```
     XTS9-143 TRANSLATE
     XTS9-143 /*
     XTS9-196 Do you want to continue the TRANSLATE process ?
     XTS9-406 ENTER 0(CANCEL) OR 1(CONTINUE)
     XTS9-403 REPLY IS 1
     XTS9-100 DATA RESTORE FEATURE VERSION 7.1.0
-->  XTS9-229 The file is a DB2 regular archive
     XTS9-193 Mount first tape of a 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 SQLDBA     .DEPARTMENT   MAY BE RELOADED
     XTS9-007 Processing successfully completed
```

Figure 160. Sample Report after Translation of a DB2 Server for VM Regular Archive

Notes:

1. The translate process reads DB2 Server for VM database archive tapes twice.
2. When the message XTS9-193 is displayed you have to remount the archive tapes beginning with the first archive tape.
3. You can restore the whole database from the translated DB2 Server for VM database archive, as well as restoring parts of the database.

UNLOAD



Purpose

The UNLOAD command unloads dbspaces to an output file.

Operands

DBSPACE=
Identify the list of dbspaces to unload.

dbspacename
Identify the names of dbspaces to process (default value for owner is PUBLIC).

* Identify that all dbspaces are included in the list. To unload a specific dbspace, enter the name of the dbspace. To unload several dbspaces at the same time, enter the list of dbspace names separated by a comma.

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

INCLUDE
Specify that the list identifies those dbspaces that are to be included in unloading. This is the default.

EXCLUDE
Specify that the list identifies those dbspaces that are to be excluded in unloading.

MODE=
Identify whether the application server is running or not during the UNLOAD process.

ONLINE
Specify that the application server is up.

OFFLINE
Specify that the application server is down.

Usage Notes

The UNLOAD command can include up to 90 dbspace names on a maximum of 10 SYSIN lines. In VSE, the REWIND parameter on the OPTIONS line, can be used to specify whether the output tape should be positioned (REWIND=YES) on not (REWIND=NO). In VM, you can use the LEAVE parameter on the FILEDEF for the

output file to achieve the same results.

Part 4. Appendixes

Appendix A. Messages and Codes

XTS9-001 Processing directory

Explanation: The Data Restore feature is processing the archive for the DB2 Server for VSE & VM directory. All directory blocks are saved.

XTS9-002 *xxx* directory blocks saved

Explanation: After the Data Restore feature has finished the directory archive, this message gives the user information about the number of blocks saved.

XTS9-005 *xxx* blocks saved

Explanation: All DBEXTENTS must be saved. Indicates how many blocks have been saved during BACKUP or UNLOAD.

XTS9-006 Processing *xxx*

Explanation: Indicates which DBEXTENT is being saved.

XTS9-007 Processing successfully completed

Explanation: Indicates that the command or exec executed successfully.

XTS9-008 Restoring directory

Explanation: The RESTORE function has been invoked, DATA RESTORE FEATURE restores the directory.

XTS9-009 *xxx* directory blocks restored

Explanation: This message displays the number of directory blocks restored.

User Response: Check to make sure the number of directory blocks is correct. If it is not correct, Contact IBM Support Center.

XTS9-010 *xxx* blocks restored

Explanation: Shows the number of DBEXTENT blocks restored.

XTS9-011 Restoring *xxx*

Explanation: The RESTORE function has been invoked, the Data Restore feature is restoring DBEXTENT xxx.

XTS9-012 12 days before password expiration

Explanation: Your password will expire in 12 days.

XTS9-013 Table *xxx.xxx* may be reloaded

Explanation: When processing BACKUP or UNLOAD function, the list of reloadable tables will be displayed on the PRINTER.

User Response: Check to make sure all tables are listed. If they are not all listed, Contact IBM Support Center.

XTS9-014 to XTS9-027

Explanation: Reports are created with these messages.

XTS9-100 The Data Restore feature Version *xxx*

Explanation: Data Restore FEATURE version is displayed on PRINTER and CONSOLE.

XTS9-101 *xxx* tables successfully processed

Explanation: With the Data Restore feature, it is possible to reload up to 90 tables in one step. At the end of the process, the number of reloaded tables is displayed.

XTS9-102 *xxx* rows loaded procedure completed

Explanation: When reloading tables the Data Restore feature the number of rows reloaded. If RESTARTCOUNT parameter has been specified, the number displayed includes SKIPPED rows.

XTS9-103 Filename (*xxx*) unknown for program (*xxx*)

Explanation: Internal error.

System Action: Processing ends.

User Response: Contact IBM Support Center.

XTS9-104 File function(*yyy*) unknown for program(*xxx*)

Explanation: Internal error. Contact IBM Support Center.

System Action: Processing ends.

User Response: Contact IBM Support Center.

XTS9-105 Invalid parameter (*yyy*) for function (*xxx*)

Explanation: The syntax of SYSIN statement is invalid or the parameter has been specified twice (refer to "OPTIONS and CONTROL Statements" on page 167 for correct syntax).

System Action: Processing ends.

User Response: Verify the syntax for the specified function and correct the SYSIN statement.

XTS9-106 Invalid command in SYSIN

Explanation: Valid commands in SYSIN are OPTIONS, CONTROL, BACKUP, RESTORE, RELOAD, FORMAT, SELECT, DESCRIBE, UNLOAD and SHOWDBS and SHOWPOOL. Correct the SYSIN statement.

System Action: Processing ends.

User Response: Verify the syntax for the specified function and correct the SYSIN statement. Verify that continuation lines start after column one.

XTS9-107 Missing parameter (*xxx*)

Explanation: A mandatory parameter is missing. Correct the SYSIN statement.

Note: If the message, XTS9-107 Missing parameter ("(" or ")" for DBSPACE=) is displayed, check that the list of dbspaces is specified on a maximum of 10 lines.

System Action: Processing ends.

User Response: Verify the syntax for the specified function and correct the SYSIN statement.

XTS9-109 Invalid value for *xxx*

Explanation: The value specified for *xxx* parameter is invalid (refer to "OPTIONS and CONTROL Statements" on page 167).

User Response: Verify the syntax for the specified function and correct the SYSIN statement.

XTS9-110 More than *xxx xxx* command(s) in SYSIN

Explanation: Only one of each command (OPTIONS, CONTROL, BACKUP, RESTORE, FORMAT, SELECT, DESCRIBE, UNLOAD, SHOWDBS, APPLYLOG, LISTLOG, TRANSLATE or SHOWPOOL can be specified in the SYSIN. Correct SYSIN statements.

System Action: Processing ends.

User Response: Verify the syntax for the specified function and correct the SYSIN statement.

XTS9-111 Length error for *xxx*

Explanation: The value for NEWCREATOR, NEWTNAME, CREATOR, TNAME, DBSPACE, OWNER, PWD, BASE, DATE, TIME or DBAPW is too long.

System Action: Processing ends.

User Response: Correct the SYSIN statement.

XTS9-113 Too many functions specified

Explanation: Only one function can be specified in SYSIN (BACKUP, RESTORE, SELECT, FORMAT, DESCRIBE, UNLOAD, SHOWDBS, APPLYLOG, LISTLOG or TRANSLATE).

System Action: Processing ends.

User Response: Correct the SYSIN statement. Verify functions are specified in column one and continuation statement after column one.

XTS9-114 No function specified

Explanation: No function (BACKUP, RESTORE, RELOAD, SELECT, FORMAT, DESCRIBE, UNLOAD or SHOWDBS) has been specified in SYSIN. Correct the SYSIN statements.

System Action: Processing ends.

User Response: Correct the SYSIN statement. Verify functions are specified in column one and continuation statement after column one.

XTS9-115 Record not found (Program=*xxx*)

Explanation: Internal error. Contact IBM Support Center.

System Action: Processing ends.

User Response: Contact IBM Support Center.

XTS9-116 More than *xxx* GETMAIN calls occurred

Explanation: Internal error. Contact IBM Support Center.

System Action: Processing ends.

XTS9-117 GETMAIN unsuccessful (R15=*xxx*)

Explanation: The WRKSIZE parameter specifies a storage size that is larger than is available in the partition (VSE) or the virtual machine (VM).

System Action: Processing ends.

User Response: Correct the parameter WRKSIZE or increase the amount of storage available for the Data Restore feature (either by running in a different partition (VSE) or increasing the size of the virtual machine (VM).

XTS9-118 FREEMAIN unsuccessful (R15=*xxx*)

Explanation: Internal error.

System Action: Processing ends.

User Response: Contact IBM Support Center.

XTS9-119 Invalid FREEMAIN address (*xxx*)

Explanation: Internal error.

System Action: Processing ends.

User Response: Contact IBM Support Center.

XTS9-120 EOF condition occurs in program *xxx*

Explanation: This message can be displayed when you specify the DBSPACE= parameter for reloading a view. Internal error. In other cases, Contact IBM Support Center.

System Action: Processing ends.

XTS9-121 Program *xxx* cannot be located

Explanation: A program is missing.

System Action: Processing ends.

System Programmer Response: Check your installation.

XTS9-122 Processing capacity exhausted(*xxx*)

Explanation: If program specified in the message is LMBRP061 or LMBRP062 increase NBVIEWS parameter on the OPTIONS statement and retry process otherwise the WRKSIZE parameter on the OPTIONS statement is too small for the Data Restore feature function.

System Action: Processing ends.

User Response: The minimum storage to specify for WRKSIZE parameter is displayed on XTS9-158 message. Modify SYSIN to specify the right WRKSIZE parameter.

XTS9-123 Invalid continuation syntax

Explanation: A continuation statement is a statement whose first word is not: CONTROL, RELOAD, BACKUP, RESTORE, FORMAT, SELECT, OPTIONS, UNLOAD, SHOWDBS, APPLYLOG, LISTLOG or TRANSLATE and begins after column 1.

User Response: Modify SYSIN to specify continuation properly.

XTS9-124 Debugger option in effect

Explanation: Internal debugger started.

XTS9-125 Debugger breakpoint *xxx*

Explanation: Specifies address of breakpoint for DATA RESTORE FEATURE internal debugger.

XTS9-126 *xxx.xxx* table cannot be located

Explanation: The table specified for RELOAD function is not found in archive, or table has been specified twice in SYSIN.

System Action: Processing ends.

User Response: Verify the TNAME and CREATOR parameters.

XTS9-127 *xxx* rows loaded, procedure continuing

Explanation: The number of loaded rows is displayed on console when the COMMITCOUNT parameter has been specified on the OPTIONS statement, and the reload process is continuing.

XTS9-128 *xxx* rows loaded into (*xxx.xxx*)

Explanation: The number of rows loaded is displayed on console when the Processing ends.

XTS9-129 SQL error (SQLCODE=*xxx*, SQLERRD1=*xxx*, SQLERRD2=*xxx*)

Explanation: An SQLERROR occurred.

System Action: Processing ends.

User Response: Refer to the *DB2 Server for VM Messages and Codes* manuals for an explanation.

XTS9-130 DBSPACE for *xxx.xxx* is unknown

Explanation: The RELOAD function tried to create a non-existent table and the parameters DBSPACE=, OWNER= are not specified on the reload SYSIN statement, or the dbspace is not acquired.

System Action: Processing ends.

User Response: Acquire the specified dbspace or specify a different value for DBSPACE= and OWNER= parameters.

XTS9-131 SQLDBA user not found or password is blank

Explanation: The RELOAD, SELECT and UNLOAD functions need an SQLDBA sign-on with DBA authority and such a sign-on is not declared in DB2 Server for VSE & VM database with a password.

System Action: Processing ends.

User Response: Execute command: GRANT DBA TO SQLDBA IDENTIFIED BY *XXXXXXXX* while connected with DBA authority.

XTS9-132 *xxx* **file** *xxx-xxx xxx* **error R15=***xxx*

Explanation: Internal error.

System Action: Processing ends.

User Response: Contact IBM Support Center.

XTS9-133 Invalid value for timestamp

Explanation: Forward log recovery requires a timestamp to process LUW up to a specific timestamp. The format is invalid.

System Action: Processing ends.

User Response: Correct the value. You can use LISTLOG function to verify the timestamp to specify.

XTS9-135 Function not supported VM/CMS)

Explanation: FORMAT function is only available for VSE environment.

System Action: Processing ends.

XTS9-136 Processing *xxx* **archived on (***xxx-xxx***)**

Explanation: Date and time of the archive is displayed.

XTS9-137 No record ″0″ found on archive

Explanation: Internal error.

System Action: Processing ends.

User Response: Verify the input file was produced by BACKUP or UNLOAD function or Contact IBM Support Center.

XTS9-138 *xxx* **archive mounted for** *xxx*

Explanation: VERIFY=YES parameter is found and archive database name is different than the base in which the table is reloaded.

System Action: Processing ends.

User Response: The database (DBNAME parameter) processed is different from the one processed during BACKUP or UNLOAD function. Execute SQLINIT with the correct DBNAME parameter. If the database is different, then specify VERIFY=NO.

XTS9-141 External labeling of this archive is

:

XTS9-142 Base *xxx* **Date** *xxx* **Time** *xxx*

Explanation: This message gives the DBNAME, date and time of the USER ARCHIVE.

XTS9-143 *xxx*

Explanation: SYSIN is displayed on PRINTER.

XTS9-145 Table *xxx.xxx* **cannot be processed**

Explanation: Table containing longvarchar columns cannot be processed by SELECT function.

System Action: Processing ends.

XTS9-146 Error at *xxx xxx xxx*

Explanation: The storage address is displayed.

System Action: Processing ends.

User Response: Verify that DBNAME parameter is correct and the server's 195 disk is accessed.

XTS9-147 File produced on *xxx* **at** *xxx* **mounted**

Explanation: Date and time of the archive files do not match the DATE= and TIME= parameters on the CONTROL STATEMENT.

System Action: Processing ends.

User Response: Mount the correct file and re-execute the process.

XTS9-148 File produced on *xxx* **at** *xxx* **required**

Explanation: Parameters DATE= and TIME= specified on CONTROL STATEMENT must correspond to archive file.

User Response: Mount the correct file or change DATE= and TIME= parameters. You can use DESCRIBE function to determine the correct values.

XTS9-150 Table *xxx.xxx* **not found**

Explanation: A SELECT function is executed for a table (CREATOR=, TNAME=) which is not found in DB2 Server for VSE & VM catalogs. TNAME must specify a table, not a view.

System Action: Processing ends.

User Response: Choose an existing table or use DESCRIBE function to determine the tables that can be processed.

XTS9-151 Column *xxx* not found in *xxx.xxx*

Explanation: A SELECT function is executed for a table, and only columns specified by COLUMNS= parameter will be displayed. One of the columns specified in COLUMNS= parameter is not found in DB2 Server for VSE & VM catalogs for the table.

System Action: Processing ends.

User Response: Change the column name or verify the table name is correct.

XTS9-152 *xxx* rows selected

Explanation: After a SELECT function, the number of rows selected is displayed.

XTS9-153 Database server message (*xxx*)

Explanation: During the Data Restore feature functions, if an SQLERROR occurs, after SQLCODE (message XTS9-129), a complementary message may be displayed.

XTS9-154 Wrong password for SQLDBA

Explanation: The SQLDBA password must be specified on the CONTROL statement when the SELECT function is used. The password specified is not valid.

System Action: Processing ends

User Response: Specify the correct password for SQLDBA user.

XTS9-155 Column *xxx* not found

Explanation: The column specified is not found in the selected table.

System Action: Processing ends.

User Response: Correct SYSIN file.

XTS9-157 Storage violation has occurred

Explanation: A Data Restore feature function has not been successful because a storage violation has occurred. A dump is created.

System Action: Processing ends.

User Response: Contact IBM Support Center.

XTS9-158 WRKSIZE needed is *xxx*

Explanation: Storage required for backup function should be at least equal to the value specified. Modify your SYSIN on OPTIONS statement specify a WRKSIZE parameter equal to or greater than this value.

System Action: Processing ends.

User Response: Modify your SYSIN to specify the required value for WRKSIZE= parameter.

XTS9-159 System Catalog tables cannot be reloaded

Explanation: The Reload function cannot process System Catalog tables. The NEWCREATOR= parameter can be specified to RELOAD System Catalog tables into a user table.

System Action: Processing ends.

User Response: Correct the SYSIN.

XTS9-160 External labeling of this unload is

Explanation: The message following this message displays date and time of the UNLOAD output file.

XTS9-161 DBSPACE *xxx.xxx* not found

Explanation: UNLOAD dbspace function cannot be executed for a non-existent dbspace, or the dbspace has been specified twice.

System Action: Processing ends.

User Response: Correct the SYSIN.

XTS9-162 SQL stmt: *xxx . xxx . xxx*

Explanation: During RELOAD process, if an SQL ERROR occurs on a dynamic statement, this message will give the program name and statement number in error and the statement will be displayed on PRINTER.

System Action: Processing ends.

User Response: Analyze SQLCODE and statement displayed or Contact IBM Support Center.

XTS9-163 Cursor name: *xxx* table: *xxx.xxx*

Explanation: During RELOAD process, if an SQL ERROR occurs, this message will give the name of the table in error and the name of the associated cursor.

System Action: Processing ends.

User Response: Analyze SQLCODE or Contact IBM Support Center.

XTS9-164 *xxx* rows skipped, procedure continuing

Explanation: During RELOAD process, if RESTARTCOUNT parameter is specified, this message will be displayed each time COMMITCOUNT parameter value is reached.

XTS9-165 *xxx* **rows loaded into (***xxx.xxx***)** *xxx* **skipped**

Explanation: During reload process, if RESTARTCOUNT parameter is specified, this message is displayed to specify how many rows were skipped and how many were loaded.

XTS9-166 **File mounted was not produced by BACKUP function**

Explanation: The RESTORE function is executed for a file that was not produced by BACKUP function.

Note: A RESTORE cannot be executed from an UNLOAD file.

System Action: Processing ends.

User Response: Mount a BACKUP file.

XTS9-167 **Record number** *xxx* **not found on LMBRWRK**

Explanation: During log recovery process a record was not found in work file but is specified in LMBRLGn files.

System Action: Processing ends.

User Response: Verify that all files were produced by the same RELOAD function.

XTS9-168 **DBSPACE** *xxx* **and Page***xxx* **not found in IXLNGVAR (step=***xxx***)**

Explanation: During log recovery for long columns, an error occurred, no matching information was found in IXLNGVAR table.

System Action: Processing ends.

User Response: Contact IBM Support Center.

XTS9-169 **Processing tables containing long columns**

Explanation: A specific process is executed for tables including long columns.

XTS9-170 **RESTARTCOUNT cannot be used with a table containing long columns**

Explanation: RESTARTCOUNT parameter can't be specified for tables having long columns.

XTS9-171 **Missing FILEDEF for** *xxx*

Explanation: A work file is used during log recovery process, the FILEDEF for this file is missing.

System Action: Processing ends.

User Response: Modify your procedure to specify the FILEDEF.

XTS9-172 **Database server was ended with logmode** *xxx*

Explanation: This message specifies the last logmode for DB2 Server for VSE & VM server. If logmode was Y, forward log recovery cannot be processed.

XTS9-173 **Database server was not ended with UARCHIVE**

Explanation: The user executes a BACKUP function but the server was not ended with the UARCHIVE parameter.

System Action: The process is stopped with RC=26.

User Response: Restart your database manager and execute command SQLEND UARCHIVE.

XTS9-174 **Processing** *xxx* **unloaded on (***xxx-xxx***)**

Explanation: The process is executed on a file generated by UNLOAD function. The DBNAME, date and time of UNLOAD function is displayed.

XTS9-175 **Recovery not allowed with file produced by UNLOAD function**

Explanation: RECOVERY=YES parameter can only be processed on a backup file.

XTS9-176 **Invalid ARCHIV file**

Explanation: The file was not produced by the BACKUP function.

System Action: Processing ends.

XTS9-177 **Recovery not allowed with file produced with logmode=***xxx*

Explanation: Recovery is allowed only for databases using LOGMODE = A or L.

System Action: Processing ends.

XTS9-178 **Recovery not allowed with logmode=***xxx*

Explanation: Recovery can only be specified for databases using log archive or archive mode.

System Action: Processing ends.

XTS9-179 **Current log**

Explanation: The current log will be required to process RELOAD function with RECOVERY=YES parameter. This message may be followed by: XTS9-236 Inactive log

XTS9-180 *xxx* **at** *xxx*

Explanation: The list of archives and log archives necessary for log recovery processing is displayed before RELOAD function starts.

XTS9-181 Archive mounted not found in history area

Explanation: The history area contains all information on date and time of archive, user archive and log archive. The user is processing a tape that is not known for database manager in history area cannot be processed.

System Action: Processing ends.

User Response: Verify the mounted archive.

XTS9-182 Following files are needed for recovery

Explanation: The list of archives and log archives necessary for log recovery processing is displayed before RELOAD function starts.

XTS9-183 Please mount log archive at *xxx*

Explanation: The log archive with the specified timestamp is required to process log recovery on reloaded tables.

User Response: Mount the required file.

XTS9-184 Processing current log

Explanation: After all log archives are processed for log recovery, the current log is processed. This message may be followed by: XTS9-235 Processing inactive log

XTS9-185 Forward recovery stopped due to *xxx*

Explanation: During forward recovery process, if a DROP TABLE, DROP DBSPACE or ALTER TABLE command on a table being reloaded is found, the process is stopped.

DROP TBL: A DROP TABLE was found
DROP DBS: A DROP DBSPACE was found
ALTER TB: An ALTER TABLE was found

System Action: Processing ends.

XTS9-186 Timestamp of statement is *xxx*

Explanation: The timestamp for the command specified in message XTS9-185 is displayed.

System Action: Processing ends.

XTS9-187 Invalid action (*xxx*) found in LMBRLG1

Explanation: The action specified in log is not INSERT, DELETE, UPDATE, DROP TABLE, DROP DBSPACE, ALTER DBSPACE.

System Action: Processing ends.

User Response: Call IBM Support Center.

XTS9-188 Wrong file mounted

Explanation: The log file specified in message XTS9-183 is not the one mounted.

System Action: Processing ends.

User Response: Verify the external file label.

XTS9-189 Timestamp *xxx* **reached in log file**

Explanation: During APPLYLOG, when END parameter is reached in log file, this message is displayed.

System Action: Processing ends.

XTS9-190 Action *xxx* **found in history area**

Explanation: Reload with log recovery cannot be processed because COLDLOG, ARCHIVE, RESTORE, STARTUP=R, SWITCH TO N, SWITCH TO Y has been executed after the archive or UARCHIVE was taken.

System Action: Processing ends.

XTS9-191 Recovery not possible

Explanation: Message XTS9-190 contains the reason why the recovery is not possible

System Action: Processing ends.

XTS9-192 Processing *xxx* **archived by database server on (***xxx-xxx***)**

Explanation: When a process accesses an archive file, the DBNAME, date and time at BACKUP or UNLOAD time is displayed.

XTS9-193 Mount first tape of database server archive

Explanation: When a database server archive is used for recovery, the archive tapes must be read twice. This message is displayed when the first tape of the archive must be mounted for the second time.

System Action: The process is waiting for operator intervention

Operator Response: Mount the first archive tape.

XTS9-194 SQLERRP(*xxx*)

Explanation: SQLERRP is displayed with SQLCODE, SQLERRD1, SQLERRD2 and SQLERRM.

User Response: Refer to the DB2 Server for VSE & VM messages and Codes manual to analyze the error.

XTS9-195 *xxx* currently mounted

Explanation: When reload is executed with RECOVERY=YES parameter, the list of all necessary files is displayed. The BACKUP file is currently mounted.

XTS9-196 Do you want to continue the *xxx* process *xxx*

Explanation: When CONFIRM=YES is specified, the user is asked to confirm that the process should continue.

XTS9-197 I/O error on LOG disk

Explanation: This message is displayed during the RELOAD process with RECOVERY=YES parameter, when the current log is processed and an I/O error occurs on disk.

System Action: If DUAL logging exists, the reload continues on the second LOG file. Otherwise, the processing is terminated.

XTS9-198 Switching to secondary log

Explanation: This message is displayed during the RELOAD process with RECOVERY=YES parameter, when the current log is processed and an I/O error occurs on disk.

System Action: If DUAL logging exists, the reload continues on the second LOG file.

XTS9-199 Dual logging not in effect, cannot switch to secondary log

Explanation: This message is displayed during the RELOAD process with RECOVERY=YES parameter, when the current log is processed and an I/O error occurs on disk.

System Action: Dual logging is not in effect. RELOAD processing is terminated.

XTS9-200 Processing larchive at *xxx*

Explanation: The log archive created at the displayed timestamp is being processed.

XTS9-201 Medium is disk *xxx xxx* *

Explanation: The log archive to be processed was created on disk. The filename and filetype are displayed.

User Response: If the log archive file was moved onto tape, the user can change the device support by replying to message XTS9-408.

XTS9-202 Processing database server archive (timestamp=*xxx*)

Explanation: The RELOAD process is executed from a DB2 Server for VSE & VM archive. The timestamp of the archive is displayed.

XTS9-203 pool=*xxx* First Extent=*xxx* Pool is *xxx*

Explanation: The SHOWPOOL function displays for all defined storpool the list of attached dbextents. This message displays the first attached dbextent. T his storpool is recoverable or non recoverable.

XTS9-204 dbextent=*xxx* Pool =*xxx* Next Dbextent=*xxx*

Explanation: The SHOWPOOL function displays for all defined storpool the list of attached dbextents. The dbextents chaining can be analyzed with this message.

XTS9-205 dbextent=*xxx* extend deleted

Explanation: The SHOWPOOL function displays for all defined storpool the list of attached dbextents. This message displays deleted extents number.

XTS9-207 Backup file used can not be restored

Explanation: Pool recovery can not process this file. The reason is describe in preceding message.

XTS9-208 the current database is not R510 or later, a pool recovery cannot be used

Explanation: The pool level recovery feature is only available for a database Version 5 Release 1 and later, the function can't be processed on the specified database.

User Response: Verify the dbname parameter in SYSIN file and control the database level.

XTS9-209 Pool number *xxx* must be included in pool list

Explanation: A pool level recovery has been executed for a list of pool. The processing has been stopped due to an error. After the problem has been solved, the list you specified in SYSIN must contain at least the same pool numbers, but you modified it and the specified

number in message has been omitted.

User Response: The specified number in message must be included in the pool list.

XTS9-210 Pool number *xxx* is not recoverable

Explanation: The storpool number specified is not recoverable. SPLR can not process it.

User Response: Remove this storpool from pool list.

XTS9-211 Beginning update of directory

Explanation: SPLR has verified that all storpool of pool list can be processed and start the update of directory. Before this message is issued user can cancel processing and restart SQL server with STARTUP=″W″. After you must let SPLR completing the process and restart SQL server with STARTUP='u'

XTS9-213 Invalid value *xxx* for pool number *xxx*

Explanation: The specified value is invalid as a pool number only defined pool numbers can be specified.

User Response: Correct the pool list to process

XTS9-214 Full restore is in progress

Explanation: If a Full restore was completed successfully, you must start DB2 Server for VSE & VM with STARTUP=U before doing something else. If a Full restore was not completed succesfully, you must run Full restore completely.

XTS9-215 Pool recovery is not allowed

Explanation: Pool recovery is not allowed after an uncompleted Full restore. This message is preceded by XTS9-214

XTS9-216 Feature not installed

:

XTS9-217 Specify FULL or INCREMENTAL on BACKUP card

Explanation: You need to specify either FULL or INCREMENTAL as the type of input file.

XTS9-218 A full backup has never been executed.

Explanation: You try to execute an Incremental Backup, but you have never executed a FULL backup. Incremental Backup is associated with a Full Backup.

System Action: You must take a full backup before trying to take an incremental backup.

XTS9-219 Mount *dbname* archived on (*date* - *time*)

Explanation: You are processing a RELOAD/RESTORE function from an INCREMENTAL backup, and are being prompted to mount the associated full backup. You must use a FULLARC FILEDEF or TLBL or DLBL to specify the file.

System Action: Mount the required file.

XTS9-220 Storage pool level recovery not allowed with INCREMENTAL BACKUP

Explanation: NEED WORDS HERE!!!

XTS9-225 Version DRF 6.1.0 required for INCREMENTAL BACKUP

Explanation: Incremental backup can only be run against Version 6 Release 1 of DB2 Server for VSE & VM.

System Action: The database must be migrated to Version 6 Release 1 before incremental backup can be used.

XTS9-226 This file is a INCREM BACKUP

Explanation: The DESCRIBE function displays this message to specify the type of the input file. In this case, the file is an INCREMENTAL backup file.

XTS9-228 The correlated full backup was archived on (*timestamp*)

Explanation: For the DESCRIBE command, this message shows the timestamp of the associated full backup for this incremental backup file.

For the RELOAD and RESTORE commands, this message is a prompt to load the associated full backup file. required.

XTS9-229 The file is a DB2 $1 archive

Explanation: The file being processed is a REGULAR|FULL archive. When processing the DESCRIBE or TRANSLATE function, this message specifies whether the input file is a REGULAR or FULL archive.

XTS9-230 WRKSIZE= $1 has been used at backup time

Explanation: To reload tables or restore a database from an incremental backup, the WRKSIZE parameter may be required if the default capacity is exhausted.

The WRKSIZE parameter to specify is equivalent to the value used at backup time.

This message displays this value.

XTS9-233 Recovery will be stopped at this point.

Explanation: During RELOAD with RECOVERY=YES a break occured in the log sequence in the history area. The operator is asked to confirm if the table should be reloaded and the log files applied until this point.

This message is prompted before reload processing.

User Response: 0 will cancel the process, 1 will continue it.

XTS9-234 Further log records will not be processed

Explanation: This message is displayed after reload processing during log analysis.

XTS9-235 Processing inactive log

Explanation: In some cases, during reload processing with RECOVERY=YES the INACTIVE log will be processed before the ACTIVE log.

XTS9-236 Inactive log

Explanation: At the beginning of reload processing with RECOVERY=YES all log files required are displayed. In some cases inactive logs will be needed.

XTS9-304 Restore from user archive invoked

:

XTS9-305 Current database will be destroyed

Explanation: Before restoring an archive, the operator is asked to confirm that the current database will be destroyed.

XTS9-307 Start the database manager with parameter "STARTUP=U"

Explanation: After restoring from a database archive, the database manager must be informed that an archive was restored. This is accomplished by specifying the STARTUP=U parameter.

XTS9-308 Format DBEXTENT invoked (*xxx*)

Explanation: Format for DBEXTENT (*xxx*) invoked.

XTS9-309 Processing DB2 for VSE and VM version *xxx*

Explanation: This message displays DB2 Server for VSE & VM version.

XTS9-311 Base *xxx* specified but archive for base *xxx* was found

Explanation: The DBNAME parameter does not match the database name recorded on the archive or unload file.

System Action: Processing ends.

User Response: Modify the DBNAME parameter to match with the mounted tape or mount another tape.

XTS9-313 DB2 for VSE and VM version *xxx* processed

Explanation: This message displays DB2 version.

XTS9-314 COMMIT WORK successful for reload of data, creating required objects

Explanation: After the table has been reloaded with FUNCT=REPLACE, a COMMIT WORK is issued and this message is displayed before the environment is created.

XTS9-315 ALL SQL statements loaded into table DATARFTR.CMD

Explanation: All commands required to recreate the environment are loaded into the DATARFTR.CMD table.

System Action: Processing ends.

User Response: If an SQLERROR occurs while recreating the environment, refer to Chapter 11, "Restoring Logical Elements" on page 125 to continue.

XTS9-401 Enter 0(STOP) or 1(CONTINUE)

Explanation: During the RELOAD process with log recovery, after each log archive is asked to be mounted this message is displayed.

System Action: Processing is waiting for operator answer.

User Response: 0 will stop the process, 1 will continue it.

XTS9-402 Enter 0 (NO) or 1 (YES)

Explanation: This message is displayed to prompt for a response to a previous message.

System Action: The system is waiting for a response.

Operator Response: Reply 0 for NO or 1 for YES.

XTS9-403 Reply is *xxx*

Explanation: This message displays the operator's response.

XTS9-405 Enter 0 (CANCEL) or 1(RETRY)

Explanation: A tape drive was not ready to allow RELOAD to begin. Reply 1 to retry the RELOAD after the tape drive is ready or reply 0 to cancel.

System Action: The system is waiting for a response.

Operator Response: Reply 0 to cancel or 1 to retry.

XTS9-406 Enter 0 (CANCEL) or 1 (CONTINUE)

Explanation: This message is displayed before the RESTORE function to confirm the destruction of the current database.

System Action: The system is waiting for a response.

Operator Response: Reply 0 to cancel or 1 to continue.

XTS9-407 Enter 0 (CANCEL), 1 (CONTINUE) or 111(SKIPFILE)

Explanation: This message is displayed during the RELOAD process, for each log archive tape, if RECOVERY=YES parameter was specified.

System Action: The system is waiting for a response.

Operator Response: Reply 0 to cancel, reply 1 to continue or reply 111 to stop.

XTS9-408 Enter 0(CANCEL), 1(CONTINUE), 11(CHANGE) or 111(SKIPFILE)

Explanation: The log archive to be processed was created on disk. The filename and filetype are displayed

System Action: The system is waiting for a response.

User Response: If the log archive file was moved to tape, the user can change the device support by replying 11. After replying 11, the operator will be asked to mount the log archive tape.

XTS9-601 DATE: *xxx* **TIME:** *xxx*

Explanation: Date and time of process is displayed.

Appendix B. Command Examples

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

Using the BACKUP Function

In VSE

```
* $$ JOB JNM=DRFBACKP,CLASS=0,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFBCKUP              BACKUP DATABASE WITH DRF TO TAPE
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
// EXEC PROC=SQLVSE02
// TLBL     ARCHIV
// ASSGN    SYS006,180
// MTC      REW,SYS006
// EXEC XTS91001,SIZE=AUTO
OPTIONS DEVICE=TAPE
CONTROL DBNAME=SQLVSE02
BACKUP
/*
/&
* $$ EOJ
```

Figure 161. JCL File for Data Restore BACKUP to Tape

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 162. 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 163. SYSIN File for Data Restore BACKUP with Dual Backup to Tape and Disk

Using the DESCRIBE Function

The following are some examples using the DESCRIBE function.

```
CONTROL BASE=S35VMDB1
DESCRIBE
```

Figure 164. SYSIN File for Data Restore DESCRIBE (DESCRIBE SYSIN)

Figure 165 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 165. EXEC File for Data Restore DESCRIBE (DESCRIBE EXEC)

Figure 166 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 VERSION 7.1.0
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 166. SYSPRINT File of Data Restore DESCRIBE (DESCRIBE SYSPRINT)

Figure 178 on page 214 shows the exec used for this function in VM, and Figure 167 on page 205 shows a VSE output sample of DESCRIBE.

```
// JOB DRFDESCR               DESCRIBE ARCHIVE TAPE
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
// 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 VERSION 7.1.0
XTS9-192 Processing SQL/VSE  archived by database manager 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 167. SYSPRINT File of Data Restore DESCRIBE

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

Figure 168. CONTROL Statement Defining an Archive for Data Restore RESTORE

If the input file is a FULL BACKUP file, message XTS9-226 will indicate that it is a FULL BACKUP file. For example:

```
        XTS9-143 CONTROL BASE=SQLDBA
        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 VERSION 7.1.0
        XTS9-136 Processing SQLDBA   archived on (02/10/97-09:03:11)
 --->   XTS9-226 This file is a FULL BACKUP
        XTS9-013 Table SQLDBA  .ACTIVITY           may be reloaded
        XTS9-013 Table SQLDBA  .DEPARTMENT         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
```

If the input file is an INCREMENTAL backup file, message XTS9-226 indicates that the file is an INCREMENTAL backup and backup file. For example:

```
       XTS9-143 CONTROL BASE=SQLDBA
       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 VERSION 7.1.0
       XTS9-136 Processing SQLDBA   archived on (02/10/97-09:03:11)
--->   XTS9-226 This file is an INCREMENTAL BACKUP
         (01/10/97-08:50:02)
       XTS9-013 Table SQLDBA  .ACTIVITY           may be reloaded
       XTS9-013 Table SQLDBA  .DEPARTMENT         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
```

The following is an example of a function report for the DESCRIBE function on a DB2 FULL archive.

```
       XTS9-143 CONTROL BASE=SQLDBA
       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 VERSION 7.1.0
       XTS9-136 Processing SQLDBA   archived on (02/10/97-09:03:11)
--->   XTS9-229 The file is a DB2 FULL archive
       XTS9-013 Table SQLDBA  .ACTIVITY           may be reloaded
       XTS9-013 Table SQLDBA  .DEPARTMENT         may be reloaded
       XTS9-007 Processing successfully completed
```

The following is an example of a function report for the DESCRIBE function on a DB2 regular archive.

```
       XTS9-143 CONTROL BASE=SQLDBA
       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 VERSION 7.1.0
       XTS9-136 Processing SQLDBA   archived on (02/10/97-09:43:25)
--->   XTS9-229 The file is a DB2 regular archive
       XTS9-013 Table SQLDBA  .ACTIVITY           may be reloaded
       XTS9-013 Table SQLDBA  .DEPARTMENT         may be reloaded
       XTS9-007 Processing successfully completed
```

Using the FORMAT Function

Figure 169 on page 207 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)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFORMAT               format data extent
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
// EXEC PROC=SQLVSE02
// EXEC XTS91001,SIZE=AUTO
OPTIONS RECOVERY=NO,DEVICE=DASD
CONTROL DBNAME=SQLVSE02
FORMAT DDSK=10
/*
/&
* $$ EOJ
```

Figure 169. JCL File for Data Restore FORMAT Dbextent

Using the RELOAD Function

Archive
The file produced by the BACKUP function, or a DB2 Server for VSE & VM database archive.

SYSIN
The file which specifies the tables to restore.

Log n
The log archive files and the current log.

Reload
The function to process.

LMBRWRK
The work file which contains pages for tables containing LONG columns.

LMBRLG1, LMBRLG2, and LMBRLG3
The files used to extract all of the changes referenced in the log files.

DB2 Server for VSE & VM database
The database directory and all dbextents.

```
XTS9-143 OPTIONS RECOVERY=YES
XTS9-143 CONTROL DBNAME=dbname
XTS9-143 RELOAD CREATOR=SQLDBA,TNAME=SUPPLIERS,FUNCT=NEW,NEWTNAME=SUPPLIERS2
XTS9-143   DBSPACE=SAMPLE
XTS9-143 /*
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-136 Processing BASE2         archived on (31/10/95-10:59:15)
XTS9-182 Following files are needed for recovery
XTS9-195 UARCHIVE          CURRENTLY MOUNTED
XTS9-180 LARCHIV        at 1995-243-11-01-05-015505
XTS9-179 CURRENT LOG
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
XTS9-403 Reply is 1
XTS9-102            1000 rows loaded procedure completed
XTS9-128            1000 rows loaded into (SQLDBA.SUPPLIERS2)
XTX9-101               1 tables successfully processed
XTS9-314 Commit work successful for reload of data, creating required objects
CONNECT SQLDBA  ;
COMMIT WORK  ;
XTS9-183 Please mount LARCHIV at 1995-243-11-01-05-015504
XTS9-401 Enter 0(stop) OR 1(continue) OR 111(skipfile)
XTS9-403 Reply is 1
XTS9-184 Processing current log
XTS9-401 Enter 0(stop) OR 1(continue) OR 111(skipfile)
XTS9-403 Reply is 1
XTS9-007 Processing successfully completed
```

Figure 170. Example Console for a RELOAD with the RECOVERY=YES Parameter

Example of a Report from RELOAD

The number of reloaded rows is displayed at the end of each RELOAD function.

```
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-136 Processing BASE1    archived on (11/10/94-15:19:11)
XTS9-128              124 Rows loaded into (SQLDBA.SUPPLIERS)
XTS9-128               25 Rows loaded into (SQLDBA.EMPLOYEE)
XTS9-101                2 Tables successfully processed
XTS9-314 Commit work successful for reload of data, creating required objects
CONNECT SQLDBA;
COMMIT WORK;
```

Figure 171. Sample Report from the RELOAD Function

If the COMMITCOUNT parameter is specified on the OPTIONS statement, a message is displayed each time a COMMIT WORK is performed. (For more information, see "OPTIONS and CONTROL Statements" on page 167.)

```
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-136 Processing BASE1    archived on (11/10/94-15:19:11)
XTS9-127               50 Rows loaded procedure  continuing
XTS9-127              100 Rows loaded procedure  continuing
XTS9-128              124 Rows loaded into (SQLDBA.SUPPLIERS)
XTX9-101                1 tables successfully processed
XTS9-314 Commit work successful for reload of data, creating required objects
CONNECT SQLDBA;
COMMIT WORK;
```

Figure 172. Example Report from the RELOAD Function with the COMMITCOUNT Parameter

In VM, if a log archive taken on DASD is subsequently moved to tape, Data Restore will still try to process the log archive from DASD. By replying **11** to message XTS9-408, you can instruct Data Restore to process the log archive from tape.

```
         XTS9-143 OPTIONS RECOVERY=YES
         XTS9-143 CONTROL  VERIFY=NO DBNAME=dbname READPW=RR
         XTS9-143 RELOAD  CREATOR=SQLDBA,TNAME=CUSTOMERS,FUNCT=REPLACE
         XTS9-143   DBSPACE=SAMPLE
         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 VERSION 7.1.0
         XTS9-136 Processing BASE620 archived on (29/09/95-13:11:37)
         XTS9-182 Following files are needed for recovery
         XTS9-195 UARCHIVE     currently mounted
         XTS9-180 LARCHIV     at 1995-272-13-13-24-714512
         XTS9-180 LARCHIV     at 1995-272-13-39-57-753488
         XTS9-179 Current log
         XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
         XTS9-403 Reply is 1
         XTS9-128       100 rows loaded into (SQLDBA.CUSTOMERS)
         XTS9-128      1481 rows loaded into (DATARFTR.SYSCOLUMNS)
         XTS9-128       317 rows loaded into (DATARFTR.SYSCATALOG)
         XTS9-128       661 rows loaded into (DATARFTR.SYSTABAUTH)
         XTS9-128       113 rows loaded into (DATARFTR.SYSINDEXES)
         XTS9-128        16 rows loaded into (DATARFTR.SYSVIEWS)
         XTS9-128        22 rows loaded into (DATARFTR.SYSKEYCOLS)
         XTS9-128        18 rows loaded into (DATARFTR.SYSKEYS)
         XTS9-128       544 rows loaded into (DATARFTR.SYSUSAGE)
         XTS9-128        28 rows loaded into (DATARFTR.SYSCOLAUTH)
         XTS9-101        10 tables successfully processed
         XTS9-314 COMMIT WORK successful for reload of data, creating required
                  objects.
         CONNECT SQLDBA  ;
         COMMIT WORK ;
(1) ---> XTS9-200 Processing LARCHIV at 1995-272-13-13-24-714512
(2) ---> XTS9-201 Medium is disk BASE35   09299502 *
(3) ---> XTS9-408 Enter 0(CANCEL),1(CONTINUE),11(CHANGE) or 111(SKIPFILE)
(4) ---> XTS9-403 Reply is 11
(5) ---> XTS9-183 Please mount LARCHIV at 1995-272-13-13-24-714512
         XTS9-407 Enter 0(CANCEL),1(CONTINUE) or 111(SKIPFILE)
(6) ---> XTS9-403 Reply is 1
(7) ---> XTS9-183 Please mount LARCHIV at 1995-272-13-39-57-753488
         XTS9-407 Enter 0(CANCEL),1(CONTINUE) or 111(SKIPFILE)
         XTS9-403 Reply is 1
(8) ---> XTS9-184 Processing current log
         XTS9-407 Enter 0(CANCEL),1(CONTINUE) or 111(SKIPFILE)
(9) ---> XTS9-403 Reply is 111
         XTS9-007 Processing successfully completed
```

Figure 173. Example Reload with the RECOVERY=YES Parameter Changing Log Archive Support

Statement 1
Displays the timestamp of the first log archive.

Statement 2
Displays the name and medium (in this case, **disk**) of the log archive.

Statement 3
Prompts the operator to change the medium or continue with the process.

Statement 4
Displays the operator's response to the above prompt (in this case, since the operator has chosen to change the medium, the new medium is assumed to be **tape**).

Statement 5
Requests that the log archive (on tape) be mounted.

Statement 6
Displays the operator's response (**continue**).

Statement 7
Requests the next log archive.

Statement 8
Indicates that the current log is being processed.

Statement 9
Displays the operator's response (not to process the current log).

RELOAD Examples using 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 174. SQLDBA.PROJECT before Updates

In Figure 175 on page 212 and Figure 176 on page 213 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 175. 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 VERSION 7.1.0
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 176. SYSPRINT File of Data Restore RELOAD from Data Restore Archive with Forward Recovery

In Figure 177 on page 214, ARCHIV is used as a work file while reading the DB2 archive (ARIARCH) from tape. The parameter ARCHTYPE=SQLDS is given in Figure 180 on page 214.

```
/*-------------------------------------------------------------------*/
/*- Reload procedure without forward recovery ----------------------*/
/*- Input=database manager 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 177. EXEC File 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 178. 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 179. EXEC File for Data Restore LISTLOG

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

Figure 180. SYSIN for Data Restore RELOAD from DB2 Archive without Forward Recovery

```
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 181. Changed Data in Table SQLDBA.PROJECT after Data Restore APPLYLOG

To reload from an INCREMENTAL backup tape, see the following example. The JCL or EXEC used for the RELOAD command must be a full backup file. The ARCHIV label or filedef should be used for the incremental backup file.

The SYSIN file used for the RELOAD command must be modified to add the WRKSIZE parameter on the OPTIONS statement. The value specified must be the same as the one specified at incremental backup time.

```
     /**/
     'TAPE REW'
     'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
     'FILEDEF LMBRWRK DISK WRK DATA A (RECFM FB BLOCK 28672
                                       LRECL 4096'
     'FILEDEF SYSPRINT DISK RESTORE SYSPRINT A'
---> 'FILEDEF FULLARC TAP1 SL 1 (RECFM VB BLOCK 32760'
     'FILEDEF SYSIN DISK RELOAD  SYSIN A'
     'XTS91001'
```

The SYSIN file should contain:

```
     OPTIONS DEVICE=TAPE DEVICE2=TAPE WRKSIZE=4096
     CONTROL DBNAME=SQLDS
     RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
     /*
```

The output from RELOAD will be:

```
     XTS9-143 OPTIONS DEVICE=TAPE DEVICE2=TAPE WRKSIZE=4096
     XTS9-143 CONTROL DBNAME=SQLDS
     XTS9-143 RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
     XTS9-143 /*
     XTS9-100 Data Restore feature VERSION 7.1.0
     XTS9-136 Processing SQLDS archived on (11/25/97-15:16:53)
---> XTS9-219 Mount BASE2 archived on (09/25/97-14:52:59)
---> XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
---> XTS9-403 Reply is 1
     XTS9-102       225 rows loaded, procedure completed
     XTS9-128       225 rows loaded into (SQLDBA.CUSTOMER)
     XTS9-101         1 tables successfully processed
```

RELOAD Examples using VSE

Figure 182 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)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB XTS9DLBL  CATALOG DLBL for data restore feature
// EXEC    LIBR,PARM='MSHP'
   ACCESS  SUBLIB=PRD2.RCVvrm
   CATALOG XTS9DLBL.PROC     EOD=&& DATA=YES REPLACE=YES
// PROC CAT='VSESPUC'
// DLBL IJSYSUC,'VSESP.USER.CATALOG',,VSAM
* ********************************************************
* XTS9DLBL: DATA RESTORE FEATURE WORKFILES
* ********************************************************
// DLBL LMBRWRK,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG1,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG2,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG3,,,VSAM,CAT=VSESPUC
// DLBL SYS0001,,0,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL HEADER,,0,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL DIRWORK,,0,VSAM,CAT=VSESPUC
&&
/*
/&
* $$ EOJ
```

Figure 182. JCL File DLBL for Data Restore Work Files (XTS9DLBL.PROC)

In Figure 183, the DLBL JCL from Figure 182 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)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFRLOAD              RELOAD TABLE WITH FORWARD RECOVERY
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
// EXEC PROC=SQLVSE02        DLBL for database extents
// EXEC PROC=XTS9DLBL        DLBL for drf files
// TLBL     ARCHIV
// ASSGN    SYS006,180       input tape from BACKUP
// MTC      REW,SYS006
// EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(SQLVSE02)'
OPTIONS RECOVERY=YES DEVICE=TAPE
CONTROL DBNAME=SQLVSE02
RELOAD CREATOR=SQLDBA TNAME=PROJECT FUNCT=REPLACE
/*
/&
* $$ EOJ
```

Figure 183. JCL File for Data Restore RELOAD from Data Restore Archive with Forward Recovery

Figure 184 on page 217 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)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFRLOAD              RELOAD TABLE WITH FORWARD RECOVERY
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
// EXEC PROC=SQLVSE02
// TLBL     ARIARCH
// ASSGN    SYS007,181             INPUT SQL ARCHIVE
// MTC      REW,SYS007
// DLBL LMBRWRK,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG1,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG2,,,VSAM,CAT=VSESPUC
// DLBL LMBRLG3,,,VSAM,CAT=VSESPUC
// DLBL SYS0001,,,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL HEADER,,,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL DIRWORK,,,VSAM,CAT=VSESPUC
// DLBL ARCHIV,,,VSAM,RECORDS=(100,100),RECSIZE=4096,CAT=VSESPUC
// EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(SQLVSE02)'
OPTIONS RECOVERY=YES,ARCHTYPE=SQLDS,DEVICE=DASD
CONTROL DBNAME=SQLVSE02
RELOAD CREATOR=SQLDBA TNAME=PROJECT FUNCT=REPLACE
/*
/&
* $$ EOJ
```

Figure 184. JCL File for Data Restore RELOAD from DB2 Archive with Forward Recovery

In Figure 185 a DB2 archive is read using SYS007 as shown in Figure 184, 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)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB SQLRLOAD    RELOAD 2 TABLEs WITH FORWARD REC from sql archi
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
// EXEC PROC=SQLVSE02
// EXEC PROC=XTS9DLBL
// TLBL     ARCHIV
// TLBL     ARIARCH
// TLBL     LARCHIV
// ASSGN    SYS006,180
// ASSGN    SYS007,181
// MTC      REW,SYS007
// EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(SQLVSE02)'
OPTIONS ARCHTYPE=SQLDS RECOVERY=YES
CONTROL DBNAME=SQLVSE02
RELOAD CREATOR=SQLDBA TNAME=PROJECT FUNCT=REPLACE
RELOAD CREATOR=SQLDBA TNAME=PROJ_ACT FUNCT=REPLACE
/*
/&
* $$ EOJ
```

Figure 185. 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)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFRLOAD    RELOAD TABLE with guest sharing
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
// TLBL     ARCHIV               * input tape *
// ASSGN    SYS006,181
// MTC      REW,SYS006
// SETPFIX  LIMIT=1M
// EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(S35VMDB1)'
OPTIONS RECOVERY=NO,DEVICE=TAPE
CONTROL DBNAME=SQLVSE02
RELOAD CREATOR=SQLDBA TNAME=ACTIVITY FUNCT=REPLACE
/*
/&
* $$ EOJ
```

Figure 186. JCL File for Data Restore RELOAD with Guest Sharing

To reload from an INCREMENTAL backup tape in, see the following example. The JCL or EXEC used for the RELOAD command must be a full backup file. The ARCHIV label or filedef should be used for the incremental backup file.

The SYSIN file used for the RELOAD command must be modified to add the WRKSIZE parameter on the OPTIONS statement. The value specified must be the same as the one specified at incremental backup time.

```
     // JOB RELOAD
     // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
     // EXEC PROC=ARIS71DB
     // DLBL LMBRWRK,,,VSAM
     // TLBL ARCHIV,'ARCHIVE.SQL',,,1
---> // TLBL FULLARC,'ARCHIVE.SQL',,,,1
     // ASSGN SYS006,180
     // MTC REW,SYS006
     // EXEC XTS91001,SIZE=AUTO
     OPTIONS DEVICE=TAPE DEVICE2=TAPE WRKSIZE=4096
     CONTROL DBNAME=SQLDS
     RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
     /*
```

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 shown in Figure 187.

```
 05.03.1997                 Control Center Facility               13.55.16
*------------------------------ Data Restore Menu -------------------------*
| Option ===> R                                              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         |
|     D  SHOWDBS                    Generate report about all dbspaces        |
|         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 187. 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 Continue
```

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 188. 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 189. 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.

```
 07.03.1997                Control Center Facility                17.24.39
*--------------------------- RELOAD EXECUTION -----------------------------*
|  DATABASE ==> ELDB2B                                    CTRLID: ELDBMSRV |
|                                                         NODE:   BOEVMCT1 |
|                                                                          |
| Execution       ==> 2            ( 1 = Schedule, 2 = Execute Immediately |
|                                                                          |
| RELOAD ID       ==> ELDB2B       (Unique file name for RELOAD event )    |
| The Reload Id (name) will be used as the file name for the RELDCTL control |
| file which will hold media and table list data required for the RELOAD.  |
| If you execute a RECOVERY=YES, the RELOAD ID is the database machine.    |
|                                                                          |
| NOTIFY ==> ________  (Userid)   ________   (Node)                        |
|                                                                          |
|                                                                          |
| Commitcount    ==> ______   ( Rows to reload before a commitcount        |
|                                                                          |
| Nbviews        ==> ______   ( Estimate number of views to recreate.     |
|                                                                          |
|                                                                          |
*-------------------------------------------------------------DRRELD5-----*
  PF1 HELP   PF3 QUIT     PF4 View RELDCTL file
                          PF7 PREVIOUS SCREEN            ENTER  CONTINUE
```

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.

Examples of Reloading Tables in a VSE Environment

The following is an example of JCL (VSE) to reload tables from a BACKUP.

Note: The Data Restore backup must be loaded on logical device SYS006. If RECOVERY=YES is specified, any log archives will also be read from logical device SYS006.

```
         // JOB RELOAD
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL LMBRWRK,,,VSAM
(3) ---> // TLBL ARCHIV,'ARCHIVE.DB2',,,1
(4) ---> // ASSGN SYS006,180
(5) ---> // MTC REW,SYS006
(6) ---> // EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(dbname)'
(7) ---> CONTROL DBNAME=dbname
(7) ---> OPTIONS DEVICE=TAPE
(7) ---> RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
(7) ---> RELOAD CREATOR=SQLDBA TNAME=SUPPLIERS
(7) --->        FUNCT=NEW
(7) --->        NEWTNAME=NEWSUPPLIERS
(7) --->        DBSPACE=SAMPLE
         /*
```

Figure 190. Sample JCL to Reload Tables from a BACKUP File

Statement 1

Specifies the DB2 Server for VSE and Data Restore libraries for the RELOAD function.

Statement 2
Specifies the work file LMBRWRK which is used if the reloaded tables contain LONG columns.

Statement 3
Specifies the ddname ARCHIV for the BACKUP file.

Statement 4
Assigns the tape containing the backup file.

Statement 5
Rewinds the tape.

Statement 6
Calls the program to execute the function. The **dbname** of the database to process must be specified on the PARM option.

Statement 7
Specifies the SYSIN for the program.

The following is an example of JCL (VSE) to reload from an UNLOAD DBSPACE.

```
         // JOB RELOAD
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL LMBRWRK,,,VSAM
(3) ---> // TLBL ARCHIV,'UNLOAD.DB2',,,1
(4) ---> // ASSGN SYS006,180
(5) ---> // MTC REW,SYS006
(6) ---> // EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(dbname)'
(7) --->      CONTROL DBNAME=dbname
              OPTIONS DEVICE=TAPE
              RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=REPLACE
         /*
```

Figure 191. Example JCL to Reload a Table from an UNLOAD File

Statement 1
Specifies the DB2 Server for VSE and Data Restore libraries for the RELOAD function.

Statement 2
Specifies the work file LMBRWRK which is used if the reloaded tables contain LONG columns.

Statement 3
Specifies the ddname ARCHIV for the UNLOAD file.

Statement 4
Assigns the tape containing the UNLOAD file.

Statement 5
Rewinds the tape.

Statement 6
Calls the program to execute the function. The **dbname** of the database to process must be specified on the PARM option.

Statement 7
Specifies the SYSIN for the program.

The following is an example of JCL (VSE) to reload and generate the log recovery file from the last BACKUP.

```
        // JOB RELOAD
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL LMBRWRK,,,VSAM
(3) ---> // DLBL LMBRLG1,,,VSAM
(3) ---> // DLBL LMBRLG2,,,VSAM
(3) ---> // DLBL LMBRLG3,,,VSAM
(4) ---> // TLBL ARCHIV,'ARCHIVE.DB2',,,1
(5) ---> // TLBL LARCHIV,,,,1
(6) ---> // ASSGN  SYS006, 180
(7) ---> // MTC REW,SYS006
(8) ---> // EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(dbname)'
(9) --->    OPTIONS RECOVERY=YES  DEVICE=TAPE
(9) --->    RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=REPLACE
        /*
```

Figure 192. Example JCL to Reload a Table and Prepare Recovery from the Log

Statement 1
: Specifies the DB2 Server for VSE & VM and Data Restore libraries for the RELOAD function.

Statement 2
: Specifies the work file LMBRWRK which is used if the reloaded tables contain LONG columns.

Statement 3
: Specifies the work files LMBRLG1, LMBRLG2, and LMBRLG3 which are used to extract all changes referenced in the log for the reloaded tables when the RECOVERY=YES parameter is specified. They will be used to execute forward log recovery.

Statement 4
: Specifies the ddname ARCHIV for the BACKUP file.

Statement 5
: Specifies the ddname LARCHIV for the log archive files.

Statement 6
: Assigns the tape containing the backup file.

Statement 7
: Rewinds the tape.

Statement 8
: Calls the program to execute the function. The **dbname** of the database to process must be specified on the PARM option.

Statement 9
: Specifies the SYSIN for the program. The OPTIONS statement specifies that log recovery is requested.

The following is an example of JCL to reload from a DB2 Server for VSE archive.

Note: The DB2 Server for VSE archive must be loaded on logical device SYS007. If a TLBL is specified for the ARCHIV file, it must be loaded on logical device SYS006. If RECOVERY=YES is specified, any log archives will also be read from logical device SYS006.

```
         // JOB RELOAD
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL LMBRWRK,,,VSAM
(3) ---> // DLBL LMBRLG1,,,VSAM
(3) ---> // DLBL LMBRLG2,,,VSAM
(3) ---> // DLBL LMBRLG3,,,VSAM
(4) ---> // DLBL ARCHIV,,,VSAM,RECSIZE=4096,RECORDS=(100,100)
(5) ---> // TLBL LARCHIV,,,,1
(6) ---> // TLBL ARIARCH,'ARCHIVE.DB2',,,1
(7) ---> // ASSGN  SYS007, 180
(8) ---> // MTC REW,SYS007
(9) ---> // DLBL SYS0001,,,VSAM,RECSIZE=4096,RECORDS=(100,100)
(9) ---> // DLBL HEADER,,,VSAM,RECSIZE=4096,RECORDS=(100,100)
(9) ---> // DLBL DIRWORK,,,VSAM
(10)---> // EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(dbname)'
(11)--->    OPTIONS  ARCHTYPE=DB2
(11)--->    CONTROL  DBNAME=dbname
(11)--->    RELOAD  CREATOR=SQLDBA,TNAME=CUSTOMERS FUNCT=NEW
(11)--->      DBSPACE=SAMPLE
         /*
```

Figure 193. Sample JCL to Execute the RELOAD Function from a DB2 Server for VSE Archiive.

Statement 1
Specifies the DB2 Server for VSE and Data Restore libraries for the RELOAD function.

Statement 2
Specifies the work file LMBRWRK which is used if the reloaded tables contain LONG columns.

Statement 3
Specifies the work files LMBRLG1, LMBRLG2, and LMBRLG3 which are used to extract all changes referenced in the log files for the reloaded tables when the RECOVERY=YES parameter is specified. They will be used to execute forward log recovery.

Statement 4
Specifies the work file ARCHIV for this process.

Statement 5
Specifies the ddname LARCHIV for the log archive file.

Statement 6
Specifies the ddname ARIARCH for the DB2 Server for VSE database archive file.

Statement 7
Assigns the tape containing the archive file.

Statement 8
Rewinds the tape.

Statement 9
Specifies the work files SYS0001, HEADER, DIRWORK for the function.

Statement 10
Calls the program to execute the function. The **dbname** of the database to process must be specified on the PARM option.

Statement 11
Specifies the SYSIN for the program. The OPTIONS statement specifies that the process is executed from a DB2 Server for VSE database archive.

The following is an example of a report generated by a RELOAD from a DB2 Server for VM archive.

```
(1) ---> XTS9-143 OPTIONS  ARCHTYPE=DB2
         XTS9-143 CONTROL  DBNAME=dbname READPW=RR
         XTS9-143 RELOAD  CREATOR=SQLDBA,TNAME=CUSTOMERS FUNCT=NEW
         XTS9-143  DBSPACE=SAMPLE
         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 VERSION 7.1.0
(2) ---> XTS9-193 Mount first tape of database server archive
         XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
         XTS9-403 Reply is 1
(3) ---> XTS9-202 Processing database server archive (timestamp=1995-274-15-50-10)
         XTS9-169 Processing tables containing long columns
         XTS9-102        20 ROWS LOADED, PROCEDURE COMPLETED
         XTS9-128        20 ROWS LOADED INTO (SQLDBA.CUSTOMERS)
         XTS9-101         1 tables successfully processed
         CONNECT SQLDBA  ;
         COMMIT WORK ;
         XTS9-007 Processing successfully completed
```

Figure 194. Example Report from a RELOAD from a DB2 Server for VSE & VM Archive

Statement 1
Specifies that the input file is a DB2 Server for VM archive.

Statement 2
Prompts the operator to remount the first archive tape.

Statement 3
Displays the timestamp of the DB2 Server for VM archive.

```
        // JOB RELOAD
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL LMBRWRK,,,VSAM
(3) ---> // TLBL ARCHIV,'ARCHIVE.SQL',,,1
(4) ---> // ASSGN SYS006,180
(5) ---> // MTC REW,SYS006
(5) ---> // EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(DBNAME)'
(7) ---> CONTROL DBNAME=DBNAME
(7) ---> OPTIONS DEVICE=TAPE
(7) ---> RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
(7) ---> RELOAD CREATOR=SQLDBA TNAME=SUPPLIERS
(7) --->        FUNCT=NEW
(7) --->        NEWTNAME=NEWSUPPLIERS
(7) --->        DBSPACE=SAMPLE
        /*
```

Figure 195. Example of JCL to Reload in a VSE Environment From a Full Backup

Statement 1 DB2 library must be specified for the RELOAD function.

Statement 2 LMBRWRK is used if the reloaded tables contain LONG columns.

Statement 3 ARCHIV is the ddname for the BACKUP file. It can be a BACKUP file or a FULL BACKUP.

Statement 4 The tape containing the backup file is assigned.

Statement 5 The tape is rewound.

Statement 6 Program to execute functions is called. The DBNAME of the database to process must be specified on the PARM option.

Statement 7 SYSIN for the program is specified.

```
          // JOB RELOAD
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL LMBRWRK,,,VSAM
(3) ---> // TLBL ARCHIV,'ARCHIVE.SQL',,,1
(4) ---> // TLBL FULLARC,'ARCHIVE.SQL',,,1
(5) ---> // ASSGN SYS006,180
(6) ---> // MTC REW,SYS006
(7) ---> // EXEC XTS91001,SIZE=AUTO,PARM='DBNAME(dbname)'
(8) ---> CONTROL DBNAME=dbname WRKSIZE=4096
(8) ---> OPTIONS DEVICE=TAPE DEVICE2=TAPE
(8) ---> RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
(8) ---> RELOAD CREATOR=SQLDBA TNAME=SUPPLIERS
(8) --->        FUNCT=NEW
(8) --->        NEWTNAME=NEWSUPPLIERS
(8) --->        DBSPACE=SAMPLE
         /*
```

Figure 196. Example of JCL to Reload in a VSE Environment From an Incremental Backup

Statement 1 DB2 library must be specified for the RELOAD function.

Statement 2 LMBRWRK is used if the reloaded tables contain LONG columns.

Statement 3 ARCHIV is the ddname for the last Incremental Backup.

Statement 4 FULLARC is the ddname for the last Full Backup.

Statement 5 The tape containing the backup file is assigned.

Statement 6 The tape is rewound.

Statement 7 Program to execute functions is called. The DBNAME of the database to process must be specified on the PARM option.

Statement 7 SYSIN for the program is specified. WRKSIZE must be equivalent to the parameter specified during the BACKUP function. DEVICE2 specifies if a TAPE or DASD (TLBL/DLBL) is used for Full Backup (FULLARC).

Examples of Reloading Tables in a VM Environment

Note: Before executing a procedure to reload tables, you **must** run the SQLINIT EXEC specifying DBNAME(**database-name**) PROTOCOL(SQLDS) to:

1. Establish **database-name** as the database into which the tables are to be reloaded.
2. Ensure that an authorization ID can be specified on a CONNECT statement.

You must also link and access the minidisk where Data Restore is installed.

The following is an example of an EXEC to RELOAD from a BACKUP or UNLOAD file.

```
/**/
'TAPE REW'
'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
'FILEDEF LMBRWRK DISK LMBRWRK DATA A (RECFM FB BLOCK 28672 LRECL 4096'
'FILEDEF SYSPRINT DISK RELOAD SYSPRINT A'
'FILEDEF SYSIN DISK RELOAD SYSIN A'
'XEDIT RELOAD SYSIN A'
'XTS91001'
```

Figure 197. Example EXEC to Execute the RELOAD Function

The SYSIN file must contain the following statements:

```
CONTROL DBNAME=dbname
RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
RELOAD CREATOR=SQLDBA TNAME=SUPPLIERS
   FUNCT=NEW
   NEWTNAME=NEWSUPPLIERS
   DBSPACE=SAMPLE
```

Figure 198. Sample SYSIN to Execute the RELOAD Function

The following is an example of an EXEC to RELOAD from the last BACKUP and generate the log recovery files.

```
(1) ---> 'TAPE REW'
(2) ---> 'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
(3) ---> 'FILEDEF LARCHIV TAP2 SL 1 (RECFM FB BLOCK 28672 LRECL 4096'
(4) ---> 'FILEDEF LMBRWRK DISK LMBRWRK DATA A (RECFM FB BLOCK 28672 LRECL 4096'
(5) ---> 'FILEDEF LMBRLG1 DISK LMBRLG1 DATA A (RECFM VB BLOCK 32760'
(5) ---> 'FILEDEF LMBRLG2 DISK LMBRLG2 DATA A (RECFM VB BLOCK 32760'
(5) ---> 'FILEDEF LMBRLG3 DISK LMBRLG3 DATA A (RECFM VB BLOCK 32760'
(6) ---> 'FILEDEF SYSPRINT DISK RELOAD SYSPRINT A'
(7) ---> 'FILEDEF SYSIN DISK RELOAD SYSIN A'
(8) ---> 'XEDIT RELOAD SYSIN A'
(9) ---> 'XTS91001'
```

Figure 199. Example EXEC to Reload Tables and Prepare Recovery from the Log

Statement 1
Rewinds the tape.

Statement 2
Specifies the ddname ARCHIV for the BACKUP file.

Statement 3
Specifies the ddname LARCHIV for the log archive file.

Statement 4
Specifies the work file LMBRWRK which is used if the reloaded tables contain LONG columns.

Statement 5
Specifies the work files LMBRLG1, LMBRLG2, and LMBRLG3 which are used to extract all changes referenced in the log for the reloaded tables when the RECOVERY=YES parameter is specified. They will be used to execute forward log recovery.

Statement 6
Specifies the SYSPRINT which contains information about the process.

Statement 7
Specifies the SYSIN file.

Statement 8
Allows you to edit the SYSIN file is make any required changes before processing.

Statement 9
Calls the program to execute the function.

The SYSIN file must contain the following statements:

```
CONTROL DBNAME=dbname
OPTIONS RECOVERY=YES
RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS
   FUNCT=REPLACE
```

Figure 200. Example SYSIN to Reload a Table and Prepare Recovery from the Log

The following is an example of an EXEC to RELOAD from a DB2 Server for VM archive.

```
         /**/
(1) ---> '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'
(2) ---> 'FILEDEF ARIARCH TAP1 SL (RECFM FB BLOCK 28672 LRECL 4096'
(3) ---> 'FILEDEF ARCHIV DISK ARCHIV DATA  G (RECFM VB BLOCK 32760'
(4) ---> 'FILEDEF LARCHIV DISK BASE2 LOGDSK1  A1(RECFM F BLOCK 4096'
(5) ---> 'FILEDEF LMBRWRK DISK LMBRWRK DATA H (RECFM FB BLOCK 28672 LRECL 4096'
(6) ---> 'FILEDEF SYS0001 DISK SYS0001 DATA H(RECFM F BLOCK 4096'
(7) ---> 'FILEDEF HEADER  DISK HEADER  DATA H(RECFM F BLOCK 4096'
(8) ---> 'FILEDEF DIRWORK DISK DIRWORK DATA H(RECFM F BLOCK 0512'
(9) ---> 'FILEDEF SYSIN DISK RELOAD   SYSIN  A'
         'FILEDEF SYSPRINT DISK RELOAD   SYSPRINT A'
         'XEDIT RELOAD SYSIN A'
         'XTS91001'
```

Figure 201. Example Procedure to Execute the RELOAD Function from a DB2 Server for VM Archive

Statement 1
Specifies the work files LMBRLG1, LMBRLG2, LMBRLG3 which are used for recovery processing.

Statement 2
Defines the virtual tape drive for DB2 Server for VM the archive file.

Statement 3
Defines a work file. This file will contain one DB2 Server for VM reloadable table.

Statement 4
Defines the database log archive file.

Statement 5
Defines the work file to process the function.

Statement 6
Defines the file which contains all the pages for dbspace SYS0001.

Statement 7
Defines the file which contains all the header pages.

Statement 8
Defines the file which contains all the directory pages.

Statement 9
Defines the SYSIN file.

The following is an example of a report generated by a RELOAD from a DB2 Server for VSE & VM archive.

```
(1) ---> XTS9-143 OPTIONS  ARCHTYPE=DB2
         XTS9-143 CONTROL  DBNAME=dbname READPW=RR
         XTS9-143 RELOAD  CREATOR=SQLDBA,TNAME=CUSTOMERS FUNCT=NEW
         XTS9-143  DBSPACE=SAMPLE
         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 VERSION 7.1.0
(2) ---> XTS9-193 Mount first tape of database server archive
         XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
         XTS9-403 Reply is 1
(3) ---> XTS9-202 Processing database server archive (timestamp=1995-274-15-50)
         XTS9-169 Processing tables containing long columns
         XTS9-102        20 ROWS LOADED, PROCEDURE COMPLETED
         XTS9-128        20 ROWS LOADED INTO (SQLDBA.CUSTOMERS)
         XTS9-101         1 tables successfully processed
         CONNECT SQLDBA  ;
         COMMIT WORK ;
         XTS9-007 Processing successfully completed
```

Figure 202. Example Report from a RELOAD from a DB2 Server for VSE & VM Archive

Statement 1
Specifies that the input file is a DB2 Server for VSE & VM archive.

Statement 2
Prompts the operator to remount the first archive tape.

Statement 3
Displays the timestamp of the DB2 Server for VSE & VM archive.

```
/**/
'TAPE REW'
'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
'FILEDEF LMBRWRK DISK LMBRWRK DATA A (RECFM FB BLOCK 28672 LRECL 4096'
'FILEDEF SYSPRINT DISK RELOAD SYSPRINT A'
'FILEDEF SYSIN DISK RELOAD SYSIN A'
'XEDIT RELOAD SYSIN A'
'XTS91001'


 SYSIN file must contain the following statements :

 CONTROL DBNAME=dbname
 RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
 RELOAD CREATOR=SQLDBA TNAME=SUPPLIERS
    FUNCT=NEW
    NEWTNAME=NEWSUPPLIERS
```

Figure 203. Example of VM Procedure to Reload From a Backup File

```
/**/
'TAPE REW'
'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
'FILEDEF FULLARC TAP1 SL1 (RECFM VB BLOCK 32760'
'FILEDEF LMBRWRK DISK LMBRWRK DATA A (RECFM FB BLOCK 28672 LRECL 4096'
'FILEDEF SYSPRINT DISK RELOAD SYSPRINT A'
'FILEDEF SYSIN DISK RELOAD SYSIN A'
'XEDIT RELOAD SYSIN A'
'XTS91001'

SYSIN file must contain the following statements :

OPTIONS WRKSIZE=4096 DEVICE2=TAPE
CONTROL DBNAME=dbname
RELOAD CREATOR=SQLDBA TNAME=CUSTOMERS FUNCT=PURGE
RELOAD CREATOR=SQLDBA TNAME=SUPPLIERS
   FUNCT=NEW
   NEWTNAME=NEWSUPPLIERS
   DBSPACE=SAMPLE
```

Figure 204. Example of VM Procedure to Reload From an Incremental Backup

In VSE

```
* $$ JOB JNM=DRFRESTR,CLASS=7,DISP=D
* $$ LST CLASS=A,DISP=D,PRI=3,DEST=(*,ELADM1)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,ELADM1)
// JOB DRFRESTR               RESTORE FULL DATABASE from tape
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCV510)
// EXEC PROC=SQLVSE02
// EXEC PROC=XTS9DLBL
// TLBL     ARCHIV
// ASSGN    SYS006,181
// MTC      REW,SYS006
/* PAUSE for monitor
// EXEC XTS91001,SIZE=AUTO
CONTROL DBNAME=SQLVSE02
RESTORE
/*
/&
* $$ EOJ
```

Figure 205. 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)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,ELADM1)
// JOB DRFRESTR               RESTORE STORAGE POOL from BACKUP tape
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCV510)
// EXEC PROC=SQLVSE02
// EXEC PROC=XTS9DLBL
// TLBL     ARCHIV
// ASSGN    SYS006,181
// MTC      REW,SYS006
/* PAUSE for monitor
// EXEC XTS91001,SIZE=AUTO
CONTROL DBNAME=SQLVSE02
RESTORE POOL=8
/*
/&
* $$ EOJ
```

Figure 206. JCL File for Data Restore RESTORE of a Storage Pool

```
// JOB DRFRESTR              DRF 4.1 RESTORE STORAGE POOL FROM BACKUP
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
// 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 7.1.0
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 207. Output File from Data Restore RESTORE of a Storage Pool

The following is an example of a RESTORE from an INCREMENTAL backup tape in VSE. The JCL or EXEC used for RESTORE must be modified to add a FULLARC label or FILEDEF is used for the incremental backup file.

The SYSIN file used for RESTORE must be modified to add the WRKSIZE parameter on the OPTIONS statement. The value specified must be the same backup time.

```
     // JOB RESTORE
     // LIBDEF *,SEARCH=(PRD2.DB2710,PRD2.RCVvrm)
     // EXEC PROC=ARIS71DB
     // TLBL ARCHIV,'ARCHIVE.SQL',,,1
---> // TLBL FULLARC,'ARCHIVE.SQL',,,,1
     // ASSGN SYS006,180
     // MTC REW,SYS006
     // EXEC XTS91001,SIZE=AUTO
     OPTIONS WRKSIZE=nnnn DEVICE2=TAPE
     CONTROL DBNAME=dbname
     RESTORE
     /*
```

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

The following is an example of a RESTORE from an INCREMENTAL backup tape in VM. The JCL or EXEC used for RESTORE must be modified to add a FULLARC label or FILEDEF is used for the incremental backup file.

The SYSIN file used for RESTORE must be modified to add the WRKSIZE parameter on the OPTIONS statement. The value specified must be the same backup time.

```
     /**/
     'TAPE REW'
     'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
     'FILEDEF SYSPRINT DISK RESTORE SYSPRINT A'
---> 'FILEDEF FULLARC TAP1 SL 1 (RECFM VB BLOCK 32760'
     'FILEDEF SYSIN DISK RESTORE SYSIN A'
     'XTS91001'
```

The SYSIN file should contain:

```
     OPTIONS WRKSIZE=nnnn DEVICE2=TAPE
     CONTROL DBNAME=dbname
     RESTORE
     /*
```

The output from RESTORE will be modified as follows:

```
      XTS9-143 OPTIONS WRKSIZE=4096 DEVICE2=TAPE
      XTS9-143 CONTROL BASE=SQLDS
      XTS9-143 RESTORE
      XTS9-143 /*
      XTS9-196 Do you want to continue the RESTORE   process ?
      XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
      XTS9-403 Reply is 1
      XTS9-100 Data Restore feature VERSION 7.1.0
      XTS9-304 Restore from user archive invoked
      XTS9-305 Current database will be destroyed
      XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
      XTS9-403 Reply is 1
      XTS9-008 Restoring directory
      XTS9-009     15661 directory blocks restored
      XTS9-011 Restoring DDSK1
      XTS9-010         7 blocks restored
      XTS9-011 Restoring DDSK3
      XTS9-010         1 blocks restored
      XTS9-011 Restoring DDSK1
      XTS9-010       199 blocks restored
      XTS9-011 Restoring DDSK3
      XTS9-010        66 blocks restored
      XTS9-011 Restoring DDSK4
--->  XTS9-219 Mount SQLDS    archived on (11/25/97-06:07:40)
--->  XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
--->  XTS9-403 Reply is 1
      XTS9-010        54 blocks restored
      XTS9-011 Restoring DDSK1
      XTS9-010        30 blocks restored
      XTS9-011 Restoring DDSK3
      XTS9-010         1 blocks restored
      XTS9-011 Restoring DDSK4
      XTS9-010        67 blocks restored
      XTS9-307 Start the database manager with parameter "STARTUP=U"
      XTS9-007 Processing successfully completed
```

Using the SELECT Function

Figure 210 on page 238 shows an example of JCL for using OUTPUT=DASD for a sequential file.

```
(1)  ---> // JOB SELECT
(2)  ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(3)  ---> // DLBL DATAUNL,'DATAUNLOAD.DATA',0
(4)  ---> // EXTENT SYS006,DOSRES,1,0,2715,300
(5)  ---> // ASSGN SYS006,DISK,VOL=DOSRES,SHR
(6)  ---> // EXEC PROC=ARIS73DB
(7)  ---> // EXEC XTS91001,SIZE=AUTO
(8)  ---> CONTROL DBAPW=XXXXXXXX
(9)  ---> SELECT CREATOR=SQLDBA TNAME=CUSTOMERS
(10) --->    OUTPUT=DASD
(11) --->    COUNT=500
(12) ---> /*
```

Figure 210. JCL to Extract Data from a Sequential File (VSE)

Statement 1
Specifies the job name.

Statement 2
Specifies the IBM DATABASE 2 Server for VSE & VM library for the SELECT function. You must specify the IBM DATABASE 2 Server for VSE & VM library for the SELECT function.

Statement 3
Defines the output VSAM file.

Statement 4-5
Defines the output file to disk.

Statement 6
Starts an EXEC to access the database files.

Statement 7
Starts the program to process the function.

Statement 8
Specifies the password for SQLDBA.

Statement 9
Indicates the creator name as well as the table name.

Statement 10
Indicates the output source (DASD).

Statement 11
Indicates the number of rows to be selected.

Statement 12
Ends the SYSIN file.

Figure 211 on page 239 shows an example of JCL for OUTPUT=DASD to a VSAM ESDS file.

```
        // JOB SELECT
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2) ---> // DLBL DATAUNL,'DATAUNLOAD.DATA',,VSAM,RECORDS=100,RECSIZE=4096
(3) ---> // EXEC PROC=ARIS73DB
(4) ---> // EXEC XTS91001,SIZE=AUTO
(5) ---> CONTROL DBAPW=XXXXXXXX
(6) ---> SELECT CREATOR=SQLDBA TNAME=CUSTOMERS
(7) --->  OUTPUT=DASD
(8) --->  COUNT=500
(9) ---> /*
```

Figure 211. JCL to Extract Data from a VSAM File (VSE)

Statement 1
Specifies the IBM DATABASE 2 Server for VSE & VM Library for the SELECT function. You must specify the IBM DATABASE 2 Server for VSE & VM library.

Statement 2
Defines the output VSAM file.

Statement 3
Uses an EXEC to access the database files.

Statement 4
Starts the program to process the function.

Statement 5
Specifies the password for SQLDBA.

Statement 6
Indicates the creator name and the table name.

Statement 7
Indicates the output device (DASD).

Statement 8
Indicates the number of rows to select.

Statement 9
Ends of SYSIN file.

Figure 212 shows an example of JCL for OUTPUT=TAPE.

```
          // JOB SELECT
(1)  ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2)  ---> // TLBL DATAUNL,'DATAUNLOAD.DATA'
(3)  ---> // ASSGN SYS006,180
(4)  ---> // MTC REW,SYS006
(5)  ---> // EXEC PROC=ARIS73DB
(6)  ---> // EXEC XTS91001,SIZE=AUTO
(7)  ---> CONTROL DBAPW=XXXXXXXX
(8)  ---> SELECT CREATOR=SQLDBA TNAME=CUSTOMERS
(9)  --->  OUTPUT=TAPE
(10) --->  /*
```

Figure 212. JCL to Extract Data from a Tape (VSE)

Statement 1
Specifies the IBM DATABASE 2 Server for VSE & VM library for the SELECT function. You must specify the IBM DATABASE 2 Server for VSE & VM library.

Statement 2-4
Defines the output file to tape.

Statement 5
Uses an EXEC to access database files

Statement 6
Starts the program to process the function.

Statement 7
Specifies the password for SQLDBA.

Statement 8
Indicates the creator name and the Table name.

Statement 9
Indicates the output device (TAPE).

Statement 10
Ends the SYSIN file.

Figure 213 shows an example of an EXEC to extract 100 rows and output them to a file on a tape.

```
        /**/
(1) ---> 'FILEDEF DATAUNL TAP1 SL 1 (RECFM VB BLOCK 32760'
(2) ---> 'FILEDEF SYSPRINT DISK SELECT SYSPRINT A'
(3) ---> 'FILEDEF SYSIN DISK SELECT SYSIN A'
(4) ---> 'XEDIT SELECT SYSIN A'
(5) ---> 'XTS91001'
```

Figure 213. Procedure to Execute a SELECT Function

Statement 1
Defines the output file on the tape device.

Statement 2
Designates the SYSPRINT file to contain information about the process.

Statement 3
Calls the SYSIN file.

Statement 4
Initiates an edit of the SYSIN file to allow for any changes before processing.

Statement 5
Starts the program to process the function.

The SYSIN file must contain the following statements:

```
(6) ---> CONTROL DBAPW=XXXXXXXX, DBNAME=dbname
(7) ---> SELECT CREATOR=SQLDBA TNAME=CUSTOMERS
(8) --->  COLUMNS=(01,02,03) OUTPUT=TAPE
(9) --->  COUNT=100
```

Figure 214. SYSIN File to Extract Data on Tape

Statement 6
Specifies the password for SQLDBA.

Statement 7
Indicates the creator name and the table name.

Statement 8
Writes the result to tape.

Statement 9
Indicates the number of ROWS to select.

Using the SHOWDBS Function

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 VERSION 7.1.0
XTS9-313 DB2 for VSE & VM Version 7 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 215. SYSPRINT File of Data Restore SHOWDBS

Using the SHOWPOOL Function

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 7.1.0
XTS9-313 DB2 for VSE & VM Version 7 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 216. SYSPRINT File of Data Restore SHOWPOOL

Using the TRANSLATE Function

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 217 on page 243).

```
 07.03.1997              Control Center Facility V5.1             11.57.18
*------------------------------- Main Menu -------------------------------*
| Option ===> u.dr                                        CTRLID: ELDBMSRV  |
|  Database => ELDB2B                                     NODE:   BOEVMCT1  |
|    ************************  DBA FUNCTIONS  ************************      |
|  O  Operator Commands                  C  Change CTRLCTR userid           |
|  S  Database Status                    P  Database Parameters             |
|  SI Database Startup (Immediate)       E  SQLEND Database (Menu)          |
|  SS Database Startup (Scheduled)       U  Database Utilities              |
|  A  Database Archiving (Menu)          R  Database Recovery (Menu)        |
|  T  Database TAPES (Menu)              M  Database Monitoring             |
|  V  View Message Log                   VJ View Database Job Schedule      |
|   ************ CONTROL CENTER ADMINISTRATOR FUNCTIONS  ***************     |
|  N  New Database Setup                 AU CTRLCTR Authorization           |
|  MS Master Schedule                    G  General CONTROL CENTER commands |
|  CCC   OO   NN  N TTTTT RRR    OO  L      CCC  EEEE NN  N TTTTT EEEE RRR  |
| CC C  O  O  NN  N   T   R  R  O  O L     CC C  E    NN  N   T   E    R  R  |
| C     O  O  N N N   T   R R   O  O L     C     E    N N N   T   E    R R   |
| C     O  O  N  NN   T   RRR   O  O L     C     EEE  N  NN   T   EEE  RRR   |
| CC C  O  O  N   N   T   R  R  O  O L     CC C  E    N   N   T   E    R  R  |
|  CCC   OO   N   N   T   R   R  OO  LLL    CCC  EEEE N   N   T   EEEE R   R |
|                                                                            |
*----------------------------------------------------------------SQMMENU------*
 PF:  1 Help   3 EXIT   5 What's New
```

Figure 217. 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 218.

```
 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       |
|     D  SHOWDBS                    Generate report about all dbspaces      |
|          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 218. 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 219. 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        |
|     D  SHOWDBS                    Generate report about all dbspaces       |
|          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 220 on page 245.

```
 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 220. Select Archive Set for Translation

From this menu (Figure 220), 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.

```
 07.03.1997                Control Center Facility                12.04.33
*------------------------- Translate Selection Set ------------------------*
| Command ==>                                              CTRLID: ELDBMSRV |
|    Database ===> ELDB2B                                  NODE:   BOEVMCT1 |
|                                                          DRMACH: ELDB2DRF |
|                                                                           |
|  Translate Set    ===> 1          Translate Set Number to Use             |
|                                   (Select from the list below)            |
|  TAPEPWD          ===> N          Specify a 'Y' (YES) for pass-           |
|                                   word authority to be handled;           |
|                                   or 'N' (NO) if READ authority           |
|                                   has already been provided.              |
|         ------------------ Valid Translate Sets -------------------       |
|  1 (07.03.1997)                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|    Enter TRANSLATE Set and press ENTER to process, or press F3 to QUIT    |
|                                                                           |
*-----------------------------------------------------------------SQMTR70-----*
 PF:   1 Help    3 End (Quit)
```

Figure 221. Translate Selection Set Menu

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

```
 07.03.1997                Control Center Facility                12.06.13
*-------------------------- Translate Verification ------------------------*
| Command ==>                                              CTRLID: ELDBMSRV |
|    Database ===> ELDB2B                                  NODE:   BOEVMCT1 |
|                                                                           |
|    This is the LAST CHANCE to CANCEL the TRANSLATE.  If you want to       |
|    continue with the TRANSLATE, press ENTER, if you want to CANCEL the    |
|    TRANSLATE now, press PF3.                                              |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|        Press ENTER to continue with the TRANSLATE request, or             |
|        press PF3 to QUIT                                                  |
|                                                                           |
*-----------------------------------------------------------------SQMAR80-----*
 PF:   1 Help    3 End (Quit)
```

Figure 222. Translate Verification

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 218 on page 243.

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 7.1.0
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
```

Figure 223. View Translate Log Using Control Center (Part 1 of 3)

```
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
```

Figure 223. View Translate Log Using Control Center (Part 2 of 3)

```
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 223. View Translate Log Using Control Center (Part 3 of 3)

For more information about the TRANSLATE operation, as well as the other Data Restore functions using Control Center, refer to the appropriate chapters of the *DB2 for VM Control Center Operations Guide*.

Refer to Figure 224 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)
* $$ PUN CLASS=A,DISP=D,PRI=3,DEST=(*,VSESQADM)
// JOB DRFTRANS               TRANSLATE SQL ARCHIVE TO DRF BACKUP
// LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm.)
// TLBL     ARIARCH
// TLBL     ARCHIV
// ASSGN    SYS007,180              input tape : database manager archive
// ASSGN    SYS006,181              output tape
// MTC      REW,SYS007
// MTC      REW,SYS006
// DLBL DIRWORK,,,VSAM,CAT=VSESPUC
// DLBL SYS0001,,,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// DLBL HEADER,,,VSAM,RECSIZE=4096,RECORDS=(100,100),CAT=VSESPUC
// EXEC XTS91001,SIZE=AUTO
TRANSLATE
/*
/&
* $$ EOJ
```

Figure 224. 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".

DB2 Archives

The following is an example of a function report for the TRANSLATE function on a DB2 FULL archive.

```
        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 7.1.0
--->    XTS9-229 The file is a DB2 FULL archive
        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 SQLDBA  .DEPARTMENT         may be reloaded
        XTS9-007 Processing successfully completed
```

The following is an example of a function report for the TRANSLATE function on a DB2 regular archive.

```
        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 7.1.0
--->    XTS9-229 The file is a DB2 regular archive
        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 SQLDBA  .DEPARTMENT         may be reloaded
        XTS9-007 Processing successfully completed
```

Using the UNLOAD Function

Figure 225 shows you the statement to unload a single dbspace.

```
(1) ---> UNLOAD DBSPACE=(dbspacename)
```

Figure 225. SYSIN File to Specify Unloading a Specific Dbspace

Statement 1

Indicates that only one dbspace (**dbspacename**) is to be unloaded.

Unloading Multiple Dbspaces

Figure 226 shows how to list up to 90 dbspaces for unloading.

```
(1) ---> UNLOAD DBSPACE =(dbspacename, dbspacename, ...)
```

Figure 226. SYSIN File to Unload More than One Dbspace

Statement 1

Indicates that a list of dbspaces will be unloaded.

Unloading All Dbspaces Except Those Listed

Figure 227 shows how to use the COND= parameter to unload all tables except for the group listed. You can have as many as 90 tables in the list.

```
(1) ---> UNLOAD DBSPACE =( dbspacename, dbspacename,...),COND=EXCLUDE
```

Figure 227. SYSIN File to Specify Excluding Dbspaces from the Unload File

Statement 1

Lists the dbspaces that are excluded from the UNLOAD process. All other dbspaces are unloaded.

If you do not specify the COND= parameter, it defaults to INCLUDE and unloads all the tables in the list.

Unloading Multiple Dbspaces with the Server Stopped (VSE)

Figure 228 shows an example of the JCL (VSE) used to unload all dbspaces except one with the application server stopped.

```
         // JOB UNLOAD
(1) ---> // LIBDEF *,SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
(2)  ---> // EXEC PROC=ARIS73DB
(3)  ---> // TLBL ARCHIV,'UNLOAD.DB2'
(4)  ---> // ASSGN SYS006,180
(5)  ---> // MTC REW,SYS006
(6)  ---> // EXEC XTS91001,SIZE=AUTO, PARM='DBNAME(dbname)'
(7)  ---> OPTIONS DEVICE=TAPE,REWIND=YES/NO
(8)  ---> CONTROL DBAPW=XXXXXXXX,DBNAME=dbname
(9)  ---> UNLOAD DBSPACE=(SAMPLE) COND=EXCLUDE MODE=OFFLINE
(10) ---> /*
```

Figure 228. JCL to Unload All Dbspaces But One Offline

Statement 1

Specifies the IBM DATABASE 2 Server for VSE & VM library for the UNLOAD function. You must specify the library.

Statement 2

Contains all DLBL defining directory, log and all dbextents.

Statement 3

Identifies the label on the tape.

Statement 4

Assigns a specific tape drive that will be used.

Statement 5

Rewinds the tape to its first file.

Statement 6

Executes the program XTS91001, pointing it to the specified **dbname**.

Statement 7

Sets the device to a tape volume.In VSE, for UNLOAD program you can specify REWIND=YES or NO,(the default value is YES). If REWIND=NO is specified at OPEN/CLOSE time the tape will not be positioned.

Statement 8

Uses the Control statement to specify the password for the SQLDBA and identifies the **dbname** to be unloaded.

Statement 9

Indicates the type of UNLOAD to perform. In this case, an UNLOAD of all dbspaces except one will be performed.

Statement 10

Ends the SYSIN file.

Unloading All Dbspaces Except One With the Server Stopped (VM)

Sample of EXEC (VM) to unload all dbspaces except one with the database server stopped.

```
        /**/
(1) ---> 'FILEDEF ARCHIV TAP1 SL 1 (RECFM VB BLOCK 32760'
(2) ---> 'FILEDEF SYSPRINT DISK UNLOAD  SYSPRINT A'
(3) ---> 'FILEDEF SYSIN DISK UNLOAD SYSIN A'
(4) ---> 'XTS91001'
```

Figure 229. Procedure to Unload All Dbspace But One Offline

The SYSIN file must contain the following statements.

```
(5) ---> CONTROL DBNAME=dbname DBAPW=XXXXXXXX
(6) ---> UNLOAD DBSPACE=(SAMPLE) COND=EXCLUDE MODE=OFFLINE
```

Figure 230. SYSIN File to Unload All Dbspaces But One Offline

Statement 1

Identifies the output file.

Statement 2

Identifies the SYSPRINT file.

Statement 3

Identifies the SYSIN file.

Statement 4

Executes the program XTS91001.

Statement 5

Uses the Control statement to specify the password for the SQLDBA and identifies the **dbname** to be unloaded.

Statement 6

Defines the type of UNLOAD to perform. In this case, an UNLOAD of all dbspaces but one will be performed.

Make sure you can access the **dbname** SQLFDEF file on the database production disk when you execute Data Restore.

Make sure you LINK and ACCESS the Data Restore minidisk to execute the UNLOAD function.

Unloading Dbspaces With the Server Up

Figure 231 on page 252 shows an example of a SYSIN file used to unload dbspaces with application server running.

```
(1) ---> UNLOAD DBSPACE =(...),COND=xxx,MODE=ONLINE
```

Figure 231. SYSIN File to Unload Dbspaces Online

Statement 1

Indicates that an UNLOAD will be performed while the application server is running. If the application server is not running an error occurs.

Make sure that you do not update dbspaces that are being unloaded while MODE=ONLINE is specified. This can result in a deadlock because of lock contention.

Because the UNLOAD function does not use SQL commands to perform the unload, the database manages the LOCK process when MODE=ONLINE is specified by issuing a LOCK DBSPACE command.

You must make sure that an an unacquired public dbspace is available when MODE=ONLINE is specified. The dbspace will be acquired (ACQUIRE PUBLIC DBSPACE command is used) and dropped (DROP DBSPACE command is used), which forces a checkpoint. The checkpoint ensures all updates to the dbspace are flushed to DASD.

Data Restore will execute a LOCK DBSPACE command in EXCLUSIVE MODE to avoid any modification on the dbspace being unloaded.

Unloading Dbspaces With the Database Server Down

Figure 232 shows an example of a SYSIN file used to unload dbspaces while the application server is down.

```
(1) ---> UNLOAD DBSPACE =(...),COND=xxx,MODE=OFFLINE
```

Figure 232. SYSIN File to Unload Dbspaces Offline

Statement 1

Indicates that the application server must be stopped before performing the unload. None of the considerations for MODE=ONLINE (Figure 231) apply here.

If the application server is running, an error occurs.

Output of an UNLOAD Function

Figure 233 on page 253 shows an example of a function report.

```
XTS9-143 CONTROL DBNAME=dbname
XTS9-143 UNLOAD  DBSPACE=(*)
XTS9-143 /*
XTS9-100 Data Restore feature VERSION 7.1.0
XTS9-309 Processing DB2 Server for VSE & VM version 7
XTS9-160 External labeling of this unload is
XTS9-142 Base BASE1    date 17/10/95 time 13:07:58
XTS9-013 Table SQLDBA.COST_TABLE may be reloaded
XTS9-013 Table SQLDBA.DEPARTMENT may be reloaded
XTS9-013 Table SQLDBA.EMPLOYEE may be reloaded
XTS9-013 Table SQLDBA.EMPLOYEE_ACTIVITY may be reloaded
XTS9-013 Table SQLDBA.FOREIG may be reloaded
XTS9-013 Table SQLDBA.INVENTORY may be reloaded
XTS9-013 Table SQLDBA.OPERATIONS may be reloaded
XTS9-013 Table SQLDBA.PLAN_TABLE may be reloaded
XTS9-013 Table SQLDBA.ROUTINE may be reloaded
XTS9-145 Table SQLDBA.STORED QUERIES may be reloaded
XTS9-013 Table SQLDBA.STRUCTURE_TABLE may be reloaded
XTS9-013 Table SQLDBA.SUPPLIERS may be reloaded
XTS9-013 Table SQLDBA.SYSTEXT2 may be reloaded
XTS9-013 Table SQLDBA.SYSUSERLIST may be reloaded
XTS9-006 Processing DDSK1
XTS9-005      1055 blocks saved
XTS9-006 Processing DDSK2
XTS9-005        78 blocks saved
XTS9-006 Processing DDSK3
XTS9-005         2 blocks saved
XTS9-007 Processing successfully completed
```

Figure 233. Function Report for the UNLOAD Function

You can use the RELOAD function to reload the tables included in an UNLOAD output file. If you do not specify an owner as part of the dbspacename, the default value is PUBLIC. To unload private dbspaces, specify DBSPACE= (**owner.dbspacename**).

You can include an OPTIONS statement to supply more specific information to Data Restore (see "OPTIONS and CONTROL Statements" on page 167).

The list of reloadable tables and the number of blocks saved is displayed on the console as shown in Figure 233.

Forward log recovery is not available when restoring from unloaded dbspaces.

Appendix C. Problem Determination

Introduction

This appendix will assist you to analyze and solve problems that may occur when using the Data Restore Feature.

The main focus of this appendix is to help the users understand and handle software errors they might encounter. Additionally, some basic information on hardware errors is also included.

This appendix helps you to:

- recognize a particular type of error
- collect and interpret the available information
- identify the actions that are necessary to remove the error.

Most error situations are indicated by a message. Therefore, messages must be read carefully. Messages may be issued by VSE, VM or DB2, or by the Data Restore .

Before any action, determine the system which issued the message.

This appendix is organized in 3 parts:

- System errors: this part helps you to analyze VM or VSE messages.
- DB2 errors: this part helps you to analyze DB2 messages.
- Data Restore Feature error: this part helps you to analyze Data Restore Feature messages.

Part 1. System Errors

For all messages issued by a system refer to the documentation for that system.

The messages are issued on SYSLOG or the system console and can be complimented with a storage DUMP.

Most of the time, during the Data Restore Feature processing, system errors will occur because hardware devices are not properly LINKed or Ready with an appropriate file mounted.

This partial list of potential problems and suggested actions will help you to solve problems before contacting your technical support.

VM Environment - Possible Problems:

1. Feature not linked to the directory and dbextents minidisks:

```
XTS9-100 Data Restore Feature Version 7.1.0
DMSXIN571I CREATING NEW FILE:
DMSXSU559W WARNING: FILE IS EMPTY
DMSXSU585E NO LINE(S) CHANGED
DMSXSU559W WARNING: FILE IS EMPTY
DMSXCT559E EMPTY FILE SQLDB2 XTS9FEDF A1 NOT WRITTEN
```

User response: Parameter DBNAME specified in SYSIN refers to the database to process. Data Restore needs to LINK the directory and dbextent minidisks to access the specified database. File **dbname** SQLFDEF defined on the production minidisk of the database server must be accessed to process the required LINK commands. Execute the required LINK and ACCESS commands to the database server production minidisk and retry the process.

2. The Server is active and access required needs Read/Write access:

```
XTS9-100 Data Restore Feature Version 7.1.0
DMSXCG517I 6 OCCURRENCE(S) CHANGED ON 6 LINE(S)
HCPLNM105E SQLDB2 0200 NOT LINKED; R/W BY SQLDB2
ARIO644E ERROR OCCURRED DURING LINK COMMAND FOR CUU 200
         THE RETURN CODE = 105.
XTS9-146 Error at EXEC XTS9FDEF DEF
```

User response: The function executed requires access in READ/WRITE mode, but the server is still active and the disk cannot be linked. The user should terminate the database server before restarting the procedure OR modify the MODE= parameter with value ONLINE to specify that the server is still active.

3. Tape device for output is not properly attached:

```
XTS9-309 Processing DB2 Server for VSE & VM version 7
DMSACC724I 200 REPLACES B (200)
DMSACC724I 200 REPLACES B (200)
DMSACC724I 201 REPLACES C (201)
DMSACC724I 201 REPLACES C (201)
XTS9-172 DB2 was ended with logmode L
DMSTLM110S ERROR READING TAP1(181)
XTS9-132 LMBRC031 file BASE2-ARCHIV OPEN ERROR R15=00
```

User response: Verify that the tape device is attach at address 181 and is ready.

4. Tape is probably write protected:

```
XTS9-100 Data Restore Feature Version 7.1.0
DMSXCG517I 6 OCCURRENCE(S) CHANGED ON 6 LINE(S)
XTS9-309 Processing DB2 Server for VSE & VM version 7
DMSACC724I 200 REPLACES B (200)
DMSACC724I 200 REPLACES B (200)
DMSACC724I 201 REPLACES C (201)
DMSACC724I 201 REPLACES C (201)
XTS9-172 DB2 was ended with logmode L
DMSTLM111S ERROR WRITING TAP1(181)
XTS9-132 LMBRC031 file BASE2-ARCHIV OPEN ERROR R15=00
```

User response: Verify that the tape mounted is not write protected and is ready.

5. Disk access is not properly defined :

```
DMSFLD069I FILEMODE H NOT ACCESSED
XTS9-100 Data Restore Feature Version 7.1.0
DMSXCG517I 6 OCCURRENCE(S) CHANGED ON 6 LINE(S)
XTS9-309 Processing DB2 Server for VSE & VM version 7
```

User response: Access all disks, where the FILEDEFs are defined.

6. Tape mounted is not a standard label tape:

```
XTS9-100 Data Restore Feature Version 7.1.0
DMSXCG517I 6 OCCURRENCE(S) CHANGED ON 6 LINE(S)
XTS9-309 Processing DB2 Server for VSE & VM version 7
DMSACC724I 200 REPLACES B (200)
DMSACC724I 200 REPLACES B (200)
DMSACC724I 201 REPLACES C (201)
DMSACC724I 201 REPLACES C (201)
XTS9-172 DB2 was ended with logmode L
DMSTLM431E TAP1(181) VOL1 LABEL MISSING
DMSTLM446R ENTER 1(VOLID) (WRITE(VOLID)), 2(REJECT) OR 3(NEWTAPE)
1(111111)
```

User response The FILEDEF specifies a standard label tape is being used, but the tape mounted has no label. Specify the valid LABEL as requested within parenthesis.

7. Error in archiving to two tapes concurrently

```
XTS9-100 Data Restore Feature Version 7.1.0
DMSXCG517I 6 OCCURRENCE(S) CHANGED ON 6 LINE(S)
XTS9-309 Processing DB2 Server for VSE & VM version 7
DMSACC724I 200 REPLACES B (200)
DMSACC724I 200 REPLACES B (200)
DMSACC724I 201 REPLACES C (201)
DMSACC724I 201 REPLACES C (201)
XTS9-172 DB2 was ended with logmode L
DMSSOP036E OPEN ERROR CODE 4 ON ARCHIV2
XTS9-132 LMBRC031 file BASE2-ARCHIV2 OPEN error R15=00
```

User response The parameter ARCHIV2 is specified to process a DUAL BACKUP but the associated FILEDEF ARCHIV2 was not defined.

Add the required FILEDEF and rerun the EXEC.

8. Error in attempting to archive to disk :

```
XTS9-100 Data Restore Feature Version 7.1.0
DMSXCG517I 6 OCCURRENCE(S) CHANGED ON 6 LINE(S)
XTS9-309 Processing DB2 Server for VSE & VM version 7
DMSACC724I 200 REPLACES B (200)
DMSACC724I 200 REPLACES B (200)
DMSACC724I 201 REPLACES C (201)
DMSACC724I 201 REPLACES C (201)
XTS9-172 DB2 was ended with logmode L
XTS9-141 External labeling of this archive is
XTS9-142 Base BASE2    Date 26/09/95 Time 10:04:15
XTS9-001 Processing directory
DMSEERD107S DISK A(191) IS FULL
DMSSCT120S OUTPUT ERROR CODE 013 ON ARCHIV2
DMSABE148T SYSTEM ABEND 001 CALLED FROM 00E3A3FC REASON CODE 000000
DMSABE141T PROGRAM INTERRUPT X'00CC' OCCURRED AT E3A3FC IN ROUTINE
CMS
```

User response: The parameter ARCHIV2 is specified to process a DUAL BACKUP, the associated FILEDEF ARCHIV2 is defined on DISK but the DISK is full. Define a larger disk or change device to tape.

9. Insufficient storage available to satisfy storage request:

```
XTS9-100 Data Restore Feature Version 7.1.0
DMSXCG517I 6 OCCURRENCE(S) CHANGED ON 6 LINE(S)
XTS9-309 Processing DB2 Server for VSE & VM version 7
DMSACC724I 200 REPLACES B (200)
DMSACC724I 200 REPLACES B (200)
DMSACC724I 201 REPLACES C (201)
DMSACC724I 201 REPLACES C (201)
XTS9-172 DB2 was ended with logmode L
XTS9-141 External labeling of this archive is
XTS9-142 Base BASE2    Date 26/09/95 Time 10:09:28
XTS9-001 Processing directory
XTS9-002      9571 directory blocks saved
DMSFR0159E INSUFFICIENT STORAGE AVAILABLE TO SATISFY FREE STORAGE REQUEST FROM
000363B8 XTS9-117 GETMAIN unsuccessful (R15=01)
READY(00016); T=1.51/3.07 10:09:48
```

User response: The CMS machine processing the function does not have enough storage to operate successfully. Define an 8 Meg virtual storage machine to process the function.

10. Error in executing the main Data Restore program:

```
backup
     7 *-* 'XTS91001'
       +++ RC(-3) +++
READY(-0003);; T=0.02/.02 10:33:45
```

User response: The CMS machine where Data Restore was installed is not linked or the XTS9GMOD procedure has not been properly processed. Link the disk, and retry the operation. If it still fails, repeat the installation process and ensure that the Data Restore Feature is properly installed.

11. Feature not properly linked:

```
XTS9-100 Data Restore Feature Version 7.1.0
DMSXCG517I 6 OCCURRENCE(S) CHANGED ON 6 LINE(S)
XTS9-309 Processing DB2 Server for VSE & VM version 7
XTS9-132 LMBRC033 file BASE2-BDISK READ error R15=08
```

User response: The disk where Data Restore was installed is likely linked with the virtual address that was used for the database directory disk.

We recommend to link the minidisk where Data Restore is installed, with an address which is not currently used by any of the application server's minidisks.

12. Feature not properly linked:

```
XTS9-100 Data Restore Feature Version 7.1.0
DMSXCG517I 6 OCCURRENCE(S) CHANGED ON 6 LINE(S)
XTS9-309 Processing DB2 Server for VSE & VM version 7
DMSACC724I 200 REPLACES B (200)
DMSACC724I 200 REPLACES B (200)
DMSACC724I 201 REPLACES C (201)
DMSACC724I 201 REPLACES C (201)
XTS9-172 DB2 was ended with logmode L
XTS9-141 External labeling of this archive is
XTS9-142 Base BASE2    Date 26/09/95 Time 10:09:28
XTS9-001 Processing directory
XTS9-002      9571 directory blocks saved
READY(00016); T=1.51/3.07 10:09:48
```

User response: The disk where The Data Restore Feature was installed is likely linked with the virtual address that was used for a dbextent.

We recommend to link the minidisk where The Data Restore Feature is installed, with an address which is not currently used by any of the application server's minidisks.

13. Failure on Restore function:

```
XTS9-100 Data Restore Feature Version 7.1.0
DMSXCG517I 6 OCCURRENCE(S) CHANGED ON 6 LINE(S)
XTS9-034 Restore from user archive invoked
XTS9-305 Current database will be destroyed
XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
1
DMSACC724I 200 REPLACES B (203)
XTS9-008 Restoring directory
DMSDK1307T FILE SYSTEM ERROR DETECTED BY DMSRCM AT ADDRESS 00E2AFBE (OFFSET
000023B6):
DMSDKD1307T WRTK REQUEST FAILED WITH CODE 8 WHILE PROCESSING FILE BASE2 BDISK B6
DMSABE2047I AUTODUMP DUMP STARTED; PLEASE WAIT
DMSABE1297I DUMP HAS BEEN TAKEN
HCPGIR450W CP ENTERED; DISABLED WAIT PSW 00020000 00F11316
```

User response: The directory size of the current database is different from the saved database (smaller). Define an appropriate size minidisk and execute a FORMAT and RESERVE for this minidisk before restarting the RESTORE procedure.

14. Incorrect password specified for minidisk access:

```
XTS9-100 Data Restore Feature Version 7.1.0
DMSXCG517I 4 OCCURRENCE(S) CHANGED ON 4 LINE(S)
ENTER WRITE PASSWORD:

HCPLNM114E SQL350 1200 NOT LINKED; MODE OR PASSWORD INCORRECT
ARI0644E An error occurred during LINK command for
         CUU 1200 .
         Return Code = 114
XTS9-146 Error at EXEC XTS9FDEF DEF
READY; T=0.20/0.36 11:45:25
```

User response: You were restoring a database but the minidisks are password protected and you entered a wrong password. You can specify this password on the CONTROL statement in SYSIN.

15. Invalid access to the server:

```
Do you want to continue the UNLOAD     process ?
XTS9-146 Enter 0(CANCEL) or 1(CONTINUE)
1
XTS9-100 Data Restore Feature Version 7.1.0
HCPLNM105E SQLDB2 0200 NOT LINKED; R/W BY SQLDB2
ARI0644E ERROR OCCURRED DURING LINK COMMAND FOR CUU 200
         THE RETURN CODE = 105.
XTS9-146 Error at EXEC XTS9FDEF DEF
READY; T=0.53/0.90 11:57:27
```

User response: The UNLOAD function was executed with OFFLINE parameter, but the server is still active. Specify ONLINE in SYSIN or terminate the server.

16. Wrong format found on tape mounted:

```
Do you want to continue the TRANSLATE process ?
XTS9-146 Enter 0(CANCEL) or 1(CONTINUE)
1
XTS9-100 Data Restore Feature Version 7.1.0
DMSSCT120S INPUT ERROR CODE 008 ON ARIARCH
DMSABE148T SYSTEM ABEND 001 CALLED FROM 00E3A3FC REASON CODE 00000000
DMSABE141T PROGRAM INTERRUPT X'00CC' OCCURRED AT E3A3FC IN ROUTINE
```

User response: The tape mounted for ARIARCH is not in the DB2 Server for VSE & VM archive format. Mount a tape produced by DB2 Server for VSE & VM standard ARCHIVE Version 7 Release 3 .

17. Incorrect National Language specification selected:

```
DMSLIO201W THE FOLLOWING NAMES ARE UNDEFINED:
 LMBRS003
READY; T=0.22/0.37 17:01:46
```

User response: The user has chosen a national language by specifying LANG= parameter. The associated TEXT was not found.

Example: LANG=S003 is specified but LMBRS003 TEXT is not found. Correct the LANG= parameter or contact IBM Support Center for assistance.

VSE Environment

1. Missing TLBL definition:

```
XTS9-100 Data Restore Feature Version 7.1.0
XTS9-309 Processing DB2 Server for VSE & VM version 7
XTS9-172 DB2 was ended with logmode Y
4183I INVALID LOGICAL UNIT  ARCHIV   SYS006
0V15I JOB LMBRBACK CANCELED. REQUEST FROM SYSTEM SERVICE ROUTINE
1S78I  JOB TERMINATED DUE TO  PROGRAM ABEND
```

   User response: The TLBL for ARCHIV file is missing. Specify a TLBL/DLBL with ddname ARCHIV.

2. A user archive is requested, but server is on-line:

```
XTS9-100 Data Restore Feature Version 7.1.0
XTS9-309 Processing DB2 Server for VSE & VM version 7
4228I FILE LOGDSK1    OPEN  ERROR X'A8'(168)
1S78I  JOB TERMINATED DUE TO  RETURN CODE
EOJ LMBRBACK  MAX.RETURN CODE=0016
```

   User response: Stop the server to process the BACKUP function.

3. Tape device not properly attached:

```
// JOB LMBRBACK
0P31A    DVC NOT OP SYSCTL=4A1
CCSW=   NOT AVAILABLE   CCB=047360
```

   User response: Verify that the tape device specified is attached properly.

4. Missing TLBL definition:

```
XTS9-100 Data Restore Feature Version 7.1.0
XTS9-309 Processing DB2 Server for VSE & VM version 7
XTS9-172 DB2 was ended with logmode Y
4881I NO LABEL INFORMATION  ARCHIV
0V15I JOB LMBRBACK CANCELED. REQUEST FROM SYSTEM SERVICE ROUTINE
```

   User response: TLBL for ARCHIV is missing. Add the required TLBL specification to the JCL.

5. LIBDEF statement points to an incorrect library :

```
// EXEC XTS91001,SIZE=XTS91001,PARM='DBNAME=dbname'
1U58I  PROGRAM NOT FOUND.
1I70I  JOB LMBRBACK CANCELED DUE TO CONTROL STATEMENT
```

User response: The LIBDEF statement is omitted or the Data Restore library is not specified. Correct the LIBDEF statement. If the problem persists, verify that Data Restore was installed correctly, and if necessary repeat the installation.

6. Incorrect specification on the Unload function:

```
XTS9-100 Data Restore Feature Version 7.1.0
XTS9-309 Processing DB2 Server for VSE & VM version 7
ARI0415D Database server &1 is not ready. Enter WAIT or CANCEL.
```

User response: An UNLOAD function is executed with MODE=ONLINE parameter, but the server is not active. Restart the server and reply WAIT on console.

7. Incorrect specification on the Unload function:

```
XTS9-100 Data Restore Feature Version 7.1.0
4228I FILE BDISK      OPEN  ERROR X'A8'(168)
1S78I  JOB TERMINATED DUE TO  RETURN CODE
EOJ LMBRBACK  MAX.RETURN CODE=0016
```

User response: An UNLOAD function is executed with MODE=OFFLINE parameter, but the server is active. Shut down the server and restart the process or specify MODE=ONLINE parameter.

8. Incorrect National Language support specification selected:

```
PROG=LMBRS003 NOT FOUND
1S55I  LAST RETURN CODE WAS 0016
```

User response: LANG=S003 parameter is specified but the associated program is not found. Correct the LANG parameter or contact your technical support.

9. Incorrect specification for the DB2 archive tape:

```
// JOB LMBRBACK
XTS9-100 Data Restore Feature Version 7.1.0
4183I INVALID LOGICAL UNIT  ARIARCH  SYS007
0V15I JOB LMBRBACK CANCELED. REQUEST FROM SYSTEM SERVICE ROUTINE
1S78I  JOB TERMINATED DUE TO  PROGRAM ABEND
EOJ LMBRBACK
```

User response: The TRANSLATE function is executing but the DB2 Server for VSE & VM archive tape was not defined with ddname ARIARCH and was not assigned to the SYS0007 address unit. Correct the JCL and retry the procedure.

10. Incorrect JCL specification for the TRANSLATE function :

```
// JOB LMBRBACK
XTS9-100 Data Restore Feature Version 7.1.0
4228I FILE DIRWORK    OPEN  ERROR X'80'(128)
1S78I  JOB TERMINATED DUE TO  RETURN CODE
EOJ LMBRBACK  MAX.RETURN CODE=0016
```

User response: The TRANSLATE function requires a DLBL for the DIRWORK, SYS0001, and HEADER work files. Correct the JCL and restart the process.

Part 2. DB2 Errors

If a DB2 error occurs while running Data Restore, the messages are displayed on SYSLOG or the system console and PRINTER. Furthermore, a dump of the DB2 command may also be produced for some errors.

Analyze the SQLCODE, SQLERRD1, and SQLERRD2 values to fix the problem. Refer to the *DB2 Server for VM Messages and Codes* manuals.

Database manager errors are listed as in Figure 234.

```
XTS9-192 Processing SQL/VSE  archived by DB2 on (26/09/95-15:20:51)
XTS9-129 DB2 error  (sqlcode=   -117,SQLERRD1=   -110,sqlerrd2=
XTS9-153 DB2 message()
XTS9-194 SQLERRP(ARIXOIN)
XTS9-162 SQL stmt:SQLDBA   .LMBREDYN.000
READY(00016); T=2.34/3.72 16:30:26
```

Figure 234. Data Restore with DB2 Error.

Both VM and VSE Environments

This partial list of potential problems and suggested actions will help you to solve problems before contacting your technical support.

1. SQLCODE = 100

   Possible solution: During the APPLYLOG function, a modification referenced in the LOG cannot be re-executed because a row has been modified between RELOAD and APPLYLOG functions. The modifications are extracted from the LOG files during the RELOAD process. The reloaded tables must not be modified before the APPLYLOG function is processed.

2. SQLCODE = - 117

   Possible solution: RELOAD PURGE was requested and the table to reload has been modified since the UNLOAD or BACKUP and now the table contains more columns than when it was saved. Execute a RELOAD with REPLACE parameter to reload the table with the correct number of columns.

3. SQLCODE = - 560

   Possible solution: The user SQLDBA has no password specified or the value specified in SYSIN file is incorrect. Correct the password and retry the process.

4. SQLCODE = -601

   Possible solution: The user specified RELOAD NEW but the table to create already exists in the database. Drop the table before re-executing the process or specify RELOAD REPLACE.

5. SQLCODE = -608

Possible solution: RELOAD function is specified with DBSPACE parameter to recreate the table in a specific dbspace, but the specified dbspace is not yet acquired. Execute an ACQUIRE DBSPACE command before re-executing the process.

6. SQLCODE = -670

   Possible solution: The reloaded table is involved with referential integrity and is a reference to other tables. The dependent tables contains keys that are not referenced in the reloaded table. For such tables, all tables must be reloaded to recreate a situation where referential integrity is consistent.

7. SQLCODE = -704

   Possible solution: An UNLOAD ONLINE function will force a checkpoint by acquiring a free dbspace. No such dbspace is available. Either an ADD DBSPACE process must be executed or an unused public dbspace should be dropped.

8. SQLCODE = -814

   Possible solution: You are unloading a system dbspace (SYS000n) with the ONLINE parameter. You cannot unload System Catalog Tables. Modify the list of dbspaces to unload.

9. SQLCODE = -940

   Possible solution: The ONLINE parameter was specified but the server is not started. Start the database server and re-execute the process.

Part 3. Data Restore Feature Errors

The Data Restore Feature messages are preceded by XTS9. This part is ordered by message number.

1. XTS9-121

   Possible solution: You have specified a national language with the LANG parameter on the OPTIONS statement (refer to "OPTIONS and CONTROL Statements" on page 167 for more information). For example, when you specify a language SSSS : program LMBRSSSS must be found. Correct the LANG parameter.

2. XTS9-122

   Possible solutions:

   If the program specified in the message is LMBRP008 : The tape was not created on a DB2 Version 7 Release 3 database. The TRANSLATE function cannot proceed. Mount the correct tape.

   If the program specified in the message is LMBRP043 : To save your database, virtual storage is required and the work size specified is too small for your database. Increase the value of the WRKSIZE parameter and retry the process.

   If the program specified in the message is LMBRP061 : You are reloading a table with full environment recreation, and the table is contains many views. Increase the NBVIEWS parameter to specify the maximum number of views to recreate and retry the process.

3. XTS9-126

   Possible solution: You are trying to RELOAD an unknown table. Verify the input file content using the DESCRIBE function or check the table name and creator in SYSIN.

4. XTS9-132

Possible solution: An open error occurred. Verify the specified file is accessible and the tape drive ready and that a FILEDEF or DLBL/TLBL is defined. Also analyze the system message on the operator console.

5. XTS9-135

   Possible solution: You are trying to execute the FORMAT function in a VM system but this function is only available for in VSE. To format a DBEXTENT in VM, you can use CMS FORMAT and RESERVE commands.

6. XTS9-137

   Possible solution: You are trying to restore a database but the input file defined with ddname ARCHIV has not been generated by a BACKUP function. If the tape was generated by a standard DB2 archive, use the TRANSLATE function to convert the archive file into a Data Restore format.

7. XTS9-145

   Possible solution: Tables containing LONGVARCHAR columns cannot be processed by the SELECT function.

8. XTS9-152

   Possible solution: You have selected a table using the SELECT function. The number of resulting rows is not the same as the one expected. The SELECT function processes pages on disk and if a checkpoint has not been issued, pages on the disk may contain old information. You can ACQUIRE and DROP a dbspace to force a checkpoint and retry your request.

9. XTS9-154

   Possible solution: An SQLDBA userid must be defined in the database with a password and have DBA authority. Verify that such a user is defined. To define an SQLDBA userid process the following command while connected with DBA authority: 'GRANT DBA TO SQLDBA IDENTIFIED BY xxxxxxxx' and retry your request.

10. XTS9-157

    Possible solution: The TRANSLATE function is processed to convert a DB2 archive file that was not created in Version 7 Release 3. Data Restore can only process DB2 Version 7 Release 3 archive files.

11. XTS9-161

    Possible solution: You are processing a dbspace with UNLOAD function but the dbspace specified does not exist. Verify the DBNAME parameter is correct. Verify the syntax for the DBSPACE parameter.

12. XTS9-166

    Possible solution: You are processing a RESTORE function, but the mounted tape was not produced by a BACKUP or TRANSLATE function. Mount the correct tape before restarting the RESTORE process.

13. XTS9-170

    Possible solution: You are reloading a table and using the RESTARTCOUNT parameter to skip a certain number of rows, but the table contains LONG columns. RESTARTCOUNT cannot be used with tables containing LONG columns. You must restart the RELOAD function without specifying RESTARTCOUNT parameter.

14. XTS9-173

    Possible solution: You are processing a BACKUP function for a database. The database server must be terminated using the SQLEND UARCHIVE operator command. The server was not stopped using this command. Restart the server to specify that command and retry the BACKUP process.

15. XTS9-190 & XTS9-191

Possible solution: You are processing a RELOAD function with forward log recovery. The database archive has been found in the log history area, but an action is found that cannot guarantee security for the recovery process (such as COLDLOG or ARCHIVE when running in LOGMODE A). Re-execute the process without RECOVERY=YES parameter.

Note: You will not be able to recover using updates in the log in this situation.

16. XTS9-309

    Possible solution: The DB2 version displayed is not correct during installation of Data Restore Feature:

    - VM - a 195 minidisk of specified version in message was linked when XTS9GMOD procedure was processed.
    - VSE- the specified library in XTS9PRDI job control was not the DB2 Version 7 Release 3 library. Re-execute this step in the installation documentation.

17. XTS9-311

    Possible solution: You are restoring from a file that was produced on a different database. Confirm to continue the process or verify that the mounted tape is the correct one.

18. RC=16

    Possible solution: If no other messages are displayed, verify that the files defined in the process (such as ARIARCH or ARCHIV) are assigned to the correct file. Retry the process after verifying the FILEDEFs (VM) or DLBLs/TLBLs (VSE).

Recover from a DB2 Server for VM Error during Full Environment Recreation

When you specify REPLACE for the FUNCT parameter during the RELOAD function, all SQL commands necessary to recreate the environment are executed. If a DB2 Server for VM error occurs during this process, the commands for environment recreation are inserted into a DB2 Server for VM table named DATARFTR.CMD and information about the SQLCODE and the program in error are displayed on SYSPRINT for VM or on the PRINTER for VSE.

When all the rows are not reloaded:

- If the COMMITCOUNT parameter was specified on the OPTIONS statement, you can use the RESTARTCOUNT parameter to skip the reloaded rows (for more information refer to "OPTIONS and CONTROL Statements" on page 167) and continue, or you can correct the problem and restart the process.

Note: If the RESTARTCOUNT parameter is specified, log recovery cannot be done.

When all the rows are reloaded:

- The commands for environment recreation are in the table DATARFTR.CMD.
- SQLCODE information and a dump of the unsuccessful command are displayed on the PRINTER. You can verify which commands were completed successfully and delete these commands from the DATARFTR.CMD table.
- Run the program XTS91002 to execute the commands in the DATARFTR.CMD table. You must specify the password for SQLDBA as a parameter.

The following is an example of JCL to run the program XTS91002 in VSE:

```
// JCL RECOVER
// LIBDEF *, SEARCH=(PRD2.DB2vrm,PRD2.RCVvrm)
// EXEC XTS91002,SIZE=AUTO,PARM='DBAPW=XXXXXXXX,DBNAME(dbname)'
/ *
```

Figure 235. Example JCL to Recover From an Environment Recreation Error

The following is an example of an EXEC to run the program XTS91002 in VM:

```
/**/
'FILEDEF SYSPRINT DISK   XTS91002  SYSPRINT A'
'XTS91002 DBAPW=XXXXXXXX'
```

Figure 236. Example EXEC to Recover From an Environment Recreation Error

The DATARFTR.CMD table has the following columns:

- SEQ, defined as SMALLINT, which contains the sequence number of each SQL statement required to recreate the environment.
- SUBSEQ, defined as SMALLINT, which contains the sequence number of each part of each SQL statement (statements longer than 60 characters have to be divided, because of the next column).
- CMD, defined as VARCHAR(60), which contains the SQL statements, split into 60 character lengths.

To select rows from the table in the proper order, you must specify **ORDER BY SEQ,SUBSEQ** on the SELECT statement.

The following is an example of rows that may be found in the DATARFTR.CMD table.

```
*
SELECT * FROM DATARFTR.CMD ORDER BY SEQ,SUBSEQ
*
SEQ    SUBSEQ     CMD
---   -------     ------------------------------------------------------------

  5       5        CREATE INDEX "SQLDBA"."ISYSTEXT1" ON "SQLDBA"."SYSTEXT1" ("T
  5      10        OPIC" ASC ) PCTFREE=10;
 10       5        COMMIT WORK ;
 15       5        CREATE UNIQUE INDEX "SQLDBA"."DEUXIEME" ON "SQLDBA"."SYSTEXT
 15      10        1" ("TOPIC" ASC ,"ITEM" ASC ) PCTFREE=10;
 20       5        COMMIT WORK ;
 25       5        CREATE INDEX "SQLDBA"."I1SYSTEXT1" ON "SQLDBA"."SYSTEXT1" ("
 25      10        TOPIC" ASC ) PCTFREE=10;
 30       5        COMMIT WORK ;
 35       5        CONNECT SQLDBA ;
 40       5        GRANT SELECT ON "SQLDBA"."SYSTEXT1" TO "PUBLIC";
 45       5        COMMIT WORK ;
 50       5        GRANT SELECT ON "SQLDBA"."SYSTEXT1" TO "GEORGE" WITH GRANT O
 50      10        PTION;
 55       5        COMMIT WORK ;
 60       5        CONNECT SQLDBA ;
 65       5        LABEL ON COLUMN "SQLDBA"."SYSTEXT1"."ITEM" IS '""';
 70       5        COMMIT WORK ;
 75       5        LABEL ON COLUMN "SQLDBA"."SYSTEXT1"."TOPIC" IS '""';
 80       5        COMMIT WORK ;
```

Figure 237. Example of Rows in DATARFTR.CMD

Reload Tables With Forward Recovery From the Log

When you reload a table from a BACKUP or a DB2 Server for VSE & VM archive file, the reloaded table does not contain any of the changes made on that table after the backup was taken. Those changes are recorded in any log archives taken since the backup and in the current log.

If you want to reload a table and apply the changes, specify RECOVERY=YES on the OPTIONS statement (refer to "OPTIONS and CONTROL Statements" on page 167) during the RELOAD process. Refer to the next sections for information on listing and applying the changes in the log.

Notes:

1. This function is available for BACKUP or DB2 Server for VSE & VM archive files processed when the server was running in LOGMODE A or L.
2. Forward Recovery can only be applied to tables that exist in the archive that is being used for the RELOAD.
3. Forward log recovery stops when it encounters a DROP TABLE or DROP DBSPACE command in the log.
4. If dual logging is being used, Data Restore will switch to the secondary log if there is an error processing the primary log.

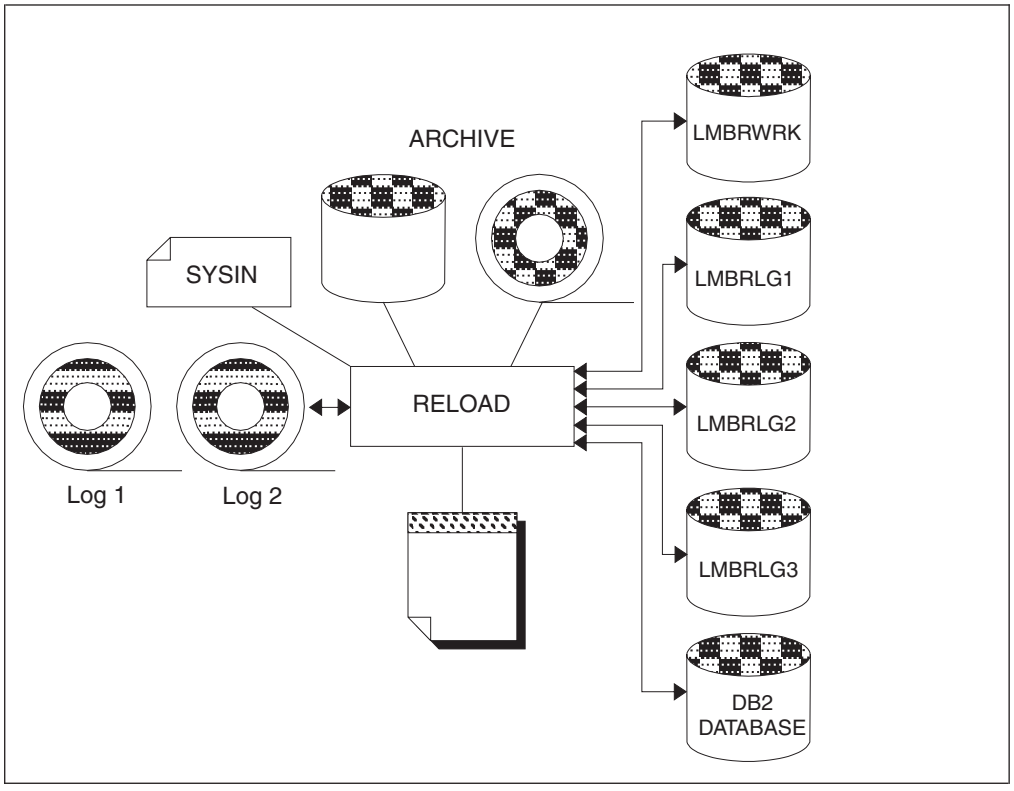


Figure 238. Possible Input Sources for RELOADing Tables with Forward Recovery

Problem Determination when Reloading Tables

If an error occurs during full environment recreation, refer to "Recover from a DB2 Server for VM Error during Full Environment Recreation" on page 266.

Problems may occur on an extended dynamic command; see Figure 239 for an example of this.

```
XTS9-143 RELOAD TNAME=SYSTEXT1
XTS9-143  CREATOR=SQLDBA
XTS9-143  FUNCT=PURGE   NEWCREATOR=GEORGE
XTS9-143 CONTROL DBNAME=dbname
XTS9-143 OPTIONS DEVICE=DASD,COMMITCOUNT=5000
XTS9-143 /*
XTS9-100 Data Restore Feature Version 7.1.0
XTS9-136 Processing SQL/VSE  archived on (14/10/94-15:38:54)
XTS9-129 SQL error (sqlcode=    -302,errd1=       0,errd2=       0)
XTS9-153 SQL msg()
XTS9-162 SQL stmt:SQLDBA  .LMBRP049.000
XTS9-163 Cursor name:C001 table:GEORGE.SYSTEXT1
```

Figure 239. DB2 Server for VSE Error on an Extended Dynamic Statement During the Reload Process

Problems may also occur on a dynamic statement; see Figure 240 for an example of this.

```
XTS9-143 RELOAD TNAME=SYSCATALOG
XTS9-143  CREATOR=DATARFTR
XTS9-143  FUNCT=NEW     NEWCREATOR=GEORGE
XTS9-143 CONTROL DBNAME=dbname
XTS9-143 OPTIONS DEVICE=DASD,COMMITCOUNT=5000
XTS9-143 /*
XTS9-100 Data Restore Feature Version 7.1.0
XTS9-136 Processing SQL/VSE  archived on (14/10/94-15:38:54)
XTS9-129 SQL error (sqlcode=    -601,errd1=       0,errd2=       0)
XTS9-153 SQL msg()
XTS9-162 SQL stmt:SQLDBA  .LMBRP017.006
```

Figure 240. DB2 Server for VSE Error on a Dynamic Statement During the Reload Process

```
LMBRP099-0209
0000  0208C3D9  C5C1E3C5  40E3C1C2  D3C5407F  D4C9D3D3  D6D57F40  4B7FE2E8  E2C3C1E3
0020  C1D3D6C7  7F4D7FE3  D5C1D4C5  7F40E5C1  D9C3C8C1  D94D40F1  F85D6B7F  C3D9C5C1
0040  E3D6D97F  40C3C8C1  D94D40F8  5D6B7FE3  C1C2D3C5  E3E8D7C5  7F40C3C8  C1D94D40
0060  F15D3B7F  D5C3D6D3  E27F40E2  D4C1D3D3  C9D5E36B  7FD9C5D4  C1D9D2E2  7F40E5C1
0080  D9C3C8C1  D94D40F2  F5F45D6B  7FC4C2E2  D7C1C3C5  D5D67F40  E2D4C1D3  D3C9D5E3
00A0  6B7FC4C2  E2D7C1C3  C5D5C1D4  C57F40E5  C1D9C3C8  C1D94D40  F1F85D6B  7FE3C1C2
00C0  C9C47F40  E2D4C1D3  6B7FC3D3  E4E2E3C5  D9E3E8D7  C57F40C3  C8C1D94D  D3C9D5E3
00E0  40F15D6B  7FC3D3E4  E2E3C5D9  D9D6E67F  40C9D5E3  C5C7C5D9  406B7FC1  E5C7D9D6
0100  E6D3C5D5  7F40E2D4  C1D3D3C9  D5E36B7F  D6E4D5E3  7F40C9D5  E3C5C7C5  D9D6E6C3
0120  D9406B7F  D5D7C1C7  C5E27F40  C9D5E3C5  C7C5D940  6B7FD7C3  E3D7C1C7  C5E27F40
0140  E2D4C1D3  D3C9D5E3  6B7FD5D6  E5C5D9C6  D3D6E67F  40C9D5E3  C5C7C5D9  406B7FD3
0160  C6C4E3C1  C2C9C47F  40E2D4C1  D3D3C9D5  E36B7FD3  C6C4D3C9  D5D27F40  E2D4C1D3
0180  D3C9D5E3  6B7FD3C6  C4C4C2E2  D7C1C3C5  7F40E2D4  C1D3D3C9  D5E36B7F  E3D3C1C2
01A0  C5D37F40  E5C1D9C3  C8C1D94D  40F3F05D  6B7FD7C1  D9C5D5E3  E27F40E2  D4C1D3D3
01C0  C9D5E36B  7FC4C5D7  C5D5C4C5  D5E3E27F  40E2D4C1  D3D3C9D5  E36B7FC9  D5C1C3E3
01E0  C9E5C57F  40E2D4C1  D3D3C9D5  E35D40C9  D5407D7   E4C2D3C9  C37F404B  7FC2C1C3
0200  D2E2E3D6  D9C57F40  40400000
```

Figure 241. Example of a Dump Produced for a DB2 Server for VSE Error on a Dynamic Statement During the Reload Process

```
*..CREATE TABLE  "GEORGE"  . "SYSCAT  *    0D2008
*ALOG" ("TNAME" VARCHAR( 18) , "CREA  *    0D2028
*TOR"  CHAR( 8) , "TABLETYPE"  CHAR ( *    0D2048
*1),"NCOLS" SMALLINT , "REMARKS"  VA  *    0D2068
*RCHAR( 254) , "DBSPACENO" SMALLINT   *    0D2088
*,"DBSPACENAME"  VARCHAR( 18) , "TAB  *    0D20A8
*ID" SMALLINT , "CLUSTERTYPE"  CHAR(  *    0D20C8
* 1) , "CLUSTERROW"  INTEGER , "AVGRO *    0D20E8
*WLEN" SMALLINT,"ROWCOUNT" INTEG      *    0D2108
*ER  , "NPAGES" INTEGER  , "PCTPAGES" *    0D2128
*SMALLINT,"NOVERFLOW" INTEGER , "L    *    0D2148
*FDTABID" SMALLINT,
LFDLINK"  SMAL   *    0D2168
*LINT, "LFDDBSPACE" SMALLINT ,"TLAB   *    0D2188
*EL"  VARCHAR ( 30) , "PARENTS"  SMALL*    0D21A8
 *INT,"DEPENDENTS" SMALLINT,"INACT    *    0D21C8
*IVE"  SMALLINT )  IN  "PUBLIC"."BAC  *    0D21E8
*KSTORE"                              0D2208
*
```

Figure 242. Example of Table Content for a DB2 Server for VSE & VM Error on a Dynamic Statement During the Reload Process

In any case, refer to the *DB2 Server for VM Messages and Codes* manual to check the meaning of the SQLCODE, or contact your database administrator or system administrator for assistance.

Note: The dump that is produced will help IBM technical support investigate the problem, if you are not able to solve it yourself.

Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10594-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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
Mail Station P300
522 South Road
Poughkeepsie, NY 12601-5400
U.S.A

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

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

CICS/VSE
DataPropagator
DB2
DFSMS
IBM
IMS
OS/390
PROFS
SQL/DS
QMF
VM/ESA
VSE/ESA

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

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

Bibliography

This bibliography lists publications that are referenced in this manual or that may be helpful.

DB2 Server for VM Publications

- *DB2 Server for VSE & VM Application Programming*, SC09-2889
- *DB2 Server for VSE & VM Database Administration*, SC09-2888
- *DB2 Server for VSE & VM Database Services Utility*, SC09-2983
- *DB2 Server for VSE & VM Diagnosis Guide and Reference*, LC09-2907
- *DB2 Server for VSE & VM Overivew*, GC09-2995
- *DB2 Server for VSE & VM Interactive SQL Guide and Reference*, SC09-2990
- *DB2 Server for VSE & VM Master Index and Glossary*, SC09-2890
- *DB2 Server for VM Messages and Codes*, GC09-2984
- *DB2 Server for VSE & VM Operation*, SC09-2986
- *DB2 Server for VSE & VM Quick Reference*, SC09-2988
- *DB2 Server for VM System Administration*, SC09-2980
- *DB2 Server for VSE & VM Performance Tuning Handbook*, GC09-2987
- *DB2 Server for VSE & VM SQL Reference*, SC09-2989

DB2 Server for VSE Publications

- *DB2 Server for VSE & VM Application Programming*, SC09-2889
- *DB2 Server for VSE & VM Database Administration*, SC09-2888
- *DB2 Server for VSE & VM Database Services Utility*, SC09-2983
- *DB2 Server for VSE & VM Diagnosis Guide and Reference*, LC09-2907
- *DB2 Server for VSE & VM Overivew*, GC09-2995
- *DB2 Server for VSE & VM Interactive SQL Guide and Reference*, SC09-2990
- *DB2 Server for VSE & VM Master Index and Glossary*, SC09-2890
- *DB2 Server for VSE Messages and Codes*, GC09-2985
- *DB2 Server for VSE & VM Operation*, SC09-2986
- *DB2 Server for VSE System Administration*, SC09-2981
- *DB2 Server for VSE & VM Performance Tuning Handbook*, GC09-2987
- *DB2 Server for VSE & VM SQL Reference*, SC09-2989

Related Publications

- *DB2 Server for VSE & VM Data Restore*, SC09-2991
- *DRDA: Every Manager's Guide*, GC26-3195
- *IBM SQL Reference, Version 2, Volume 1*, SC26-8416
- *IBM SQL Reference*, SC26-8415

VM/ESA Publications

- *VM/ESA: General Information*, GC24-5745
- *VM/ESA: VMSES/E Introduction and Reference*, GC24-5837
- *VM/ESA: Installation Guide*, GC24-5836
- *VM/ESA: Service Guide*, GC24-5838
- *VM/ESA: Planning and Administration*, SC24-5750
- *VM/ESA: CMS File Pool Planning, Administration, and Operation*, SC24-5751
- *VM/ESA: REXX/EXEC Migration Tool for VM/ESA*, GC24-5752
- *VM/ESA: Conversion Guide and Notebook*, GC24-5839
- *VM/ESA: Running Guest Operating Systems*, SC24-5755
- *VM/ESA: Connectivity Planning, Administration, and Operation*, SC24-5756
- *VM/ESA: Group Control System*, SC24-5757
- *VM/ESA: System Operation*, SC24-5758
- *VM/ESA: Virtual Machine Operation*, SC24-5759
- *VM/ESA: CP Programming Services*, SC24-5760
- *VM/ESA: CMS Application Development Guide*, SC24-5761
- *VM/ESA: CMS Application Development Reference*, SC24-5762
- *VM/ESA: CMS Application Development Guide for Assembler*, SC24-5763
- *VM/ESA: CMS Application Development Reference for Assembler*, SC24-5764

- *VM/ESA: CMS Application Multitasking*, SC24-5766
- *VM/ESA: CP Command and Utility Reference*, SC24-5773
- *VM/ESA: CMS Primer*, SC24-5458
- *VM/ESA: CMS User's Guide*, SC24-5775
- *VM/ESA: CMS Command Reference*, SC24-5776
- *VM/ESA: CMS Pipelines User's Guide*, SC24-5777
- *VM/ESA: CMS Pipelines Reference*, SC24-5778
- *VM/ESA: XEDIT User's Guide*, SC24-5779
- *VM/ESA: XEDIT Command and Macro Reference*, SC24-5780
- *VM/ESA: Quick Reference*, SX24-5290
- *VM/ESA: Performance*, SC24-5782
- *VM/ESA: Dump Viewing Facility*, GC24-5853
- *VM/ESA: System Messages and Codes*, GC24-5841
- *VM/ESA: Diagnosis Guide*, GC24-5854
- *VM/ESA: CP Diagnosis Reference*, SC24-5855
- *VM/ESA: CP Diagnosis Reference Summary*, SX24-5292
- *VM/ESA: CMS Diagnosis Reference*, SC24-5857
- CP and CMS control block information is not provided in book form. This information is available on the IBM VM/ESA operating system home page (http://www.ibm.com/s390/vm).
- *IBM VM/ESA: CP Exit Customization*, SC24-5672
- *VM/ESA REXX/VM User's Guide*, SC24-5465
- *VM/ESA REXX/VM Reference*, SC24-5770

C for VM/ESA Publications

- *IBM C for VM/ESA Diagnosis Guide*, SC09-2149
- *IBM C for VM/ESA Language Reference*, SC09-2153
- *IBM C for VM/ESA Compiler and Run-Time Migration Guide*, SC09-2147
- *IBM C for VM/ESA Programming Guide*, SC09-2151
- *IBM C for VM/ESA User's Guide*, SC09-2152

Virtual Storage Extended/Enterprise Systems Architecture (VSE/ESA) Publications

- *IBM VSE/ESA Administration*, SC33-6505
- *IBM VSE/ESA Diagnosis Tools*, SC33-6514
- *IBM VSE/ESA General Information*, GC33-6501
- *IBM VSE/ESA Guide for Solving Problems*, SC33-6510
- *IBM VSE/ESA Guide to System Functions*, SC33-6511
- *IBM VSE/ESA Installation*, SC33-6504
- *IBM VSE/ESA Messages & Codes*, SC33-6507
- *IBM VSE/ESA Networking Support*, SC33-6508
- *IBM VSE/ESA Operation*, SC33-6506
- *IBM VSE/ESA Planning*, SC33-6503
- *IBM VSE/ESA System Control Statements*, SC33-6513
- *IBM VSE/ESA System Macros User's Guide*, SC33-6515
- *IBM VSE/ESA System Macros Reference*, SC33-6516
- *IBM VSE/ESA System Utilities*, SC33-6517
- *IBM VSE/ESA Unattended Node Support*, SC33-6512
- *IBM VSE/ESA Using IBM Workstations*, SC33-6509

CICS/VSE Publications

- *CICS/VSE Application Programming Reference*, SC33-0713
- *CICS/VSE Application Programming Guide*, SC33-0712
- *CICS Application Programming Primer (VS COBOL II)*, SC33-0674
- *CICS/VSE CICS-Supplied Transactions*, SC33-0710
- *CICS/VSE Customization Guide*, SC33-0707
- *CICS/VSE Facilities and Planning Guide*, SC33-0718
- *CICS/VSE Intercommunication Guide*, SC33-0701
- *CICS/VSE Performance Guide*, SC33-0703
- *CICS/VSE Problem Determination Guide*, SC33-0716
- *CICS/VSE Recovery and Restart Guide*, SC33-0702
- *CICS/VSE Release Guide*, GC33-1645
- *CICS/VSE Report Controller User's Guide*, SC33-0705
- *CICS/VSE Resource Definition (Macro)*, SC33-0709
- *CICS/VSE Resource Definition (Online)*, SC33-0708
- *CICS/VSE System Definition and Operations Guide*, SC33-0706
- *CICS/VSE System Programming Reference*, SC33-0711
- *CICS/VSE User's Handbook*, SX33-6079
- *CICS/VSE XRF Guide*, SC33-0704

CICS/ESA Publications

- *CICS/ESA General Information*, GC33-0803

VSE/Virtual Storage Access Method (VSE/VSAM) Publications

- *VSE/VSAM Commands and Macros*, SC33-6532
- *VSE/VSAM Introduction*, GC33-6531
- *VSE/VSAM Messages and Codes*, SC24-5146
- *VSE/VSAM Programmer's Reference*, SC33-6535

VSE/Interactive Computing and Control Facility (VSE/ICCF) Publications

- *VSE/ICCF Administration and Operation*, SC33-6562
- *VSE/ICCF Primer*, SC33-6561
- *VSE/ICCF User's Guide*, SC33-6563

VSE/POWER Publications

- *VSE/POWER Administration and Operation*, SC33-6571
- *VSE/POWER Application Programming*, SC33-6574
- *VSE/POWER Networking*, SC33-6573
- *VSE/POWER Remote Job Entry*, SC33-6572

Distributed Relational Database Architecture (DRDA) Library

- *Application Programming Guide*, SC26-4773
- *Architecture Reference*, SC26-4651
- *Connectivity Guide*, SC26-4783
- *DRDA: Every Manager's Guide*, GC26-3195
- *Planning for Distributed Relational Database*, SC26-4650
- *Problem Determination Guide*, SC26-4782

C/370 for VSE Publications

- *IBM C/370 General Information*, GC09-1386
- *IBM C/370 Programming Guide for VSE*, SC09-1399
- *IBM C/370 Installation and Customization Guide for VSE*, GC09-1417
- *IBM C/370 Reference Summary for VSE*, SX09-1246
- *IBM C/370 Diagnosis Guide and Reference for VSE*, LY09-1805

VSE/REXX Publication

- *VSE/REXX Reference*, SC33-6642

Other Distributed Data Publications

- *IBM Distributed Data Management (DDM) Architecture, Architecture Reference, Level 4*, SC21-9526
- *IBM Distributed Data Management (DDM) Architecture, Implementation Programmer's Guide*, SC21-9529
- *VM/Directory Maintenance Licensed Program Specification*, GC20-1836
- *IBM Distributed Relational Database Architecture Reference*, SC26-4651
- *IBM Systems Network Architecture, Format and Protocol Reference*, SC30-3112
- *SNA LU 6.2 Reference: Peer Protocols*, SC31-6808
- *Reference Manual: Architecture Logic for LU Type 6.2*, SC30-3269
- *IBM Systems Network Architecture, Logical Unit 6.2 Reference: Peer Protocols*, SC31-6808
- *Distributed Data Management (DDM) General Information*, GC21-9527

CCSID Publications

- *Character Data Representation Architecture, Executive Overview*, GC09-2207
- *Character Data Representation Architecture Reference and Registry*, SC09-2190

DB2 Server RXSQL Publications

- *DB2 REXX SQL for VM/ESA Installation and Reference*, SC09-2891

C/370 Publications

- *IBM C/370 Installation and Customization Guide*, GC09-1387
- *IBM C/370 Programming Guide*, SC09-1384

Communication Server for OS/2 Publications

- *Up and Running!*, GC31-8189
- *Network Administration and Subsystem Management Guide*, SC31-8181
- *Command Reference*, SC31-8183
- *Message Reference*, SC31-8185
- *Problem Determination Guide*, SC31-8186

Distributed Database Connection Services (DDCS) Publications

- *DDCS User's Guide for Common Servers*, S20H-4793
- *DDCS for OS/2 Installation and Configuration Guide*, S20H-4795

VTAM Publications

- *VTAM Messages and Codes*, SC31-6493
- *VTAM Network Implementation Guide*, SC31-6494
- *VTAM Operation*, SC31-6495
- *VTAM Programming*, SC31-6496
- *VTAM Programming for LU 6.2*, SC31-6497
- *VTAM Resource Definition Reference*, SC31-6498
- *VTAM Resource Definition Samples*, SC31-6499

CSP/AD and CSP/AE Publications

- *Developing Applications*, SH20-6435
- *CSP/AD and CSP/AE Installation Planning Guide*, GH20-6764
- *Administering CSP/AD and CSP/AE on VM*, SH20-6766
- *Administering CSP/AD and CSP/AE on VSE*, SH20-6767
- *CSP/AD and CSP/AE Planning*, SH20-6770
- *Cross System Product General Information*, GH23-0500

Query Management Facility (QMF) Publications

- *Introducing QMF*, GC27-0714
- *Installing and Managing QMF for VSE*, GC27-0721
- *QMF Reference*, SC27-0715
- *Installing and Managing QMF for VM*, GC27-0720
- *Developing QMF Applications*, SC27-0718
- *QMF Messages and Codes*, GC27-0717
- *Using QMF*, SC27-0716

Query Management Facility (QMF) for Windows Publications

- *Getting Started with QMF for Windows*, SC27-0723
- *Installing and Managing QMF for Windows*, GC27-0722

DL/I DOS/VS Publications

- *DL/I DOS/VS Application Programming*, SH24-5009

COBOL Publications

- *VS COBOL II Migration Guide for VSE*, GC26-3150
- *VS COBOL II Migration Guide for MVS and CMS*, GC26-3151
- *VS COBOL II General Information*, GC26-4042
- *VS COBOL II Language Reference*, GC26-4047
- *VS COBOL II Application Programming Guide*, SC26-4045
- *VS COBOL II Application Programming Debugging*, SC26-4049
- *VS COBOL II Installation and Customization for CMS*, SC26-4213
- *VS COBOL II Installation and Customization for VSE*, SC26-4696
- *VS COBOL II Application Programming Guide for VSE*, SC26-4697

Data Facility Storage Management Subsystem/VM (DFSMS/VM) Publications

- *DFSMS/VM RMS User's Guide and Reference*, SC35-0141

Systems Network Architecture (SNA) Publications

- *SNA Transaction Programmer's Reference Manual for LU Type 6.2*, GC30-3084
- *SNA Format and Protocol Reference: Architecture Logic for LU Type 6.2*, SC30-3269
- *SNA LU 6.2 Reference: Peer Protocols*, SC31-6808
- *SNA Synch Point Services Architecture Reference*, SC31-8134

Miscellaneous Publications

- *IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide*, GA32-0100
- *Dictionary of Computing*, ZC20-1699
- *APL2 Programming: Using Structured Query Language*, SH21-1056
- *ESA/390 Principles of Operation*, SA22-7201

Related Feature Publications

- *DB2 for VM Control Center Operations Guide*, GC09-2993
- *DB2 for VSE Control Center Operations Guide*, GC09-2992
- *DB2 Replication Guide and Reference*, SC26-9920

Index

G

H

I

L

M

N

O

P

R

S

T

U

V

Contacting IBM

Before you contact DB2 customer support, check the product manuals for help with your specific technical problem.

For information or to order any of the DB2 Server for VSE & VM products, contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-237-5511 for customer support
- 1-888-426-4343 to learn about available service options

Product information

DB2 Server for VSE & VM product information is available by telephone or by the World Wide Web at http://www.ibm.com/software/data/db2/vse-vm

This site contains the latest information on the technical library, product manuals, newsgroups, APARs, news, and links to web resources.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-IBM-CALL (1-800-426-2255) to order products or to obtain general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, go to the IBM Worldwide page at http://www.ibm.com/planetwide

In some countries, IBM-authorized dealers should contact their dealer support structure for information.

File Number: S370/4300-50
Program Number: 5697-F42

Printed in U.S.A.

SC09-2991-01

Spine information:

DB2 Server for VSE & VM **Data Restore Guide** Version 7 Release 3